

LOOK SHARP: AMIGA GRAPHICS!

Amazing

COMPUTING™

Your Original AMIGA® Monthly Resource

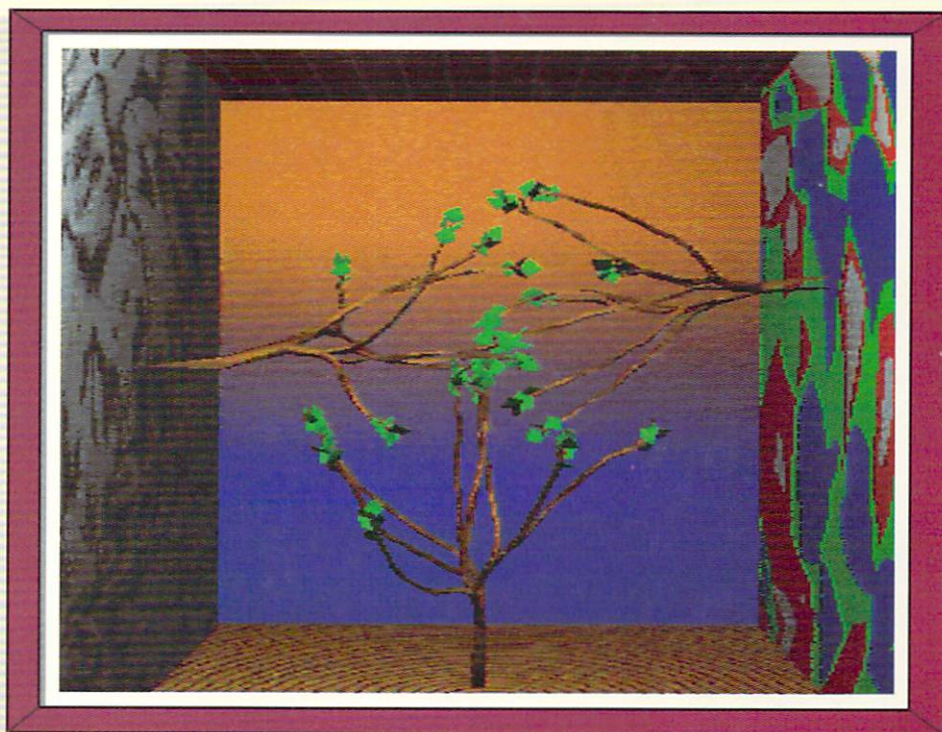
For The Commodore

AMIGA®

Volume 5 No. 9 September 1990
US \$3.95 Canada \$4.95

GRAPHICS

3-D Professional
Scene Generator
Frame Grabber
Color Dithering
And More!



David Duberman's
Textures

Plus

Dr. T's KCS 3.0

Programming In C On A Floppy System

Building A Voice-Controlled Joystick



CONTENTS

GRAPHICS

INSIDE STORIES

KCS 3.0 Review 10

by Phil Saunders
Dr. T's Keyboard Controlled Sequencer has significant changes.

Acting on Impulse 31

by John Steiner
Impulse's Firecracker 24 board breaks the 4096-color barrier.

3-D Professional 46

by David Duberman
Progressive Peripherals' 3-D modeling, rendering, and animation program.

How Pro is 3-D Professional? 61

by Frank McMahon
Simply put, it makes 3-D output a breeze.

Programming in C on a Floppy System 66

by Paul Miller
Yes, even on a stock Amiga 500 with a 512K RAM expander.



On the cover: David Duberman's graphic image created with Progressive Peripherals & Software's 3-D Professional. Accompanying article can be found on page 46.

PROGRAMMING

Time Out! 50

by Mark Cashman
Accessing the Amiga's system timer device via Modula-2.

Stock-Portfolio 72

by G. L. Penrose
Here's an original program to organize your investments, music library, mailing lists, and more!

CygCC 81

by Duncan Thompson
An ARexx programming tutorial.

REVIEWS

The Game of Harmony 63

by Joe DiCara
A game of challenge that rewards a relaxed approach and a calm state of mind.

Future Wars: Adventures in Time 63

by Miguel Mulet
Can you save humanity?

DEPARTMENTS

Editorial 4

Feedback 6

List of Advertisers 80

Public Domain
Software 91

COLUMNS

New Products and Other Neat Stuff 15

compiled by Elizabeth G. Fedorzyn

A high-density drive from Applied Engineering, enhancements to some major Commodore products, and more.

Snapshot 55

by R. Brad Andrews

Race your Porsche 959 from New York to California in Sega's Turbo Outrun.



MICROPROSE'S Pirates game.

Voice-Controlled Joystick 58

by John Iovine

Build your own voice-controlled joystick!

Bug Bytes 69

by John Steiner

Gold Disk's Transcript, M2Sprint, and a new technical support person at Intuitive Technologies.

Roomers 77

by The Bandito

Nintendo's Super Famicom is headed for Japan. And whatever happened to the Amiga 250?

C Notes From

The C Group 87

by Stephen Kemp

Plan your C programs to take advantage of the Amiga's multitasking capabilities.

GRAPHICS

FrameGrabber 18

by Frank McMahon

"...this package is miles ahead... in ease of use, logic, and professional image enhancement."

KARA Fonts 23

by R. Shamms Mortier

KARA Computer Graphics' KARA Fonts collection is here to stay.

Gradient Color Dithering on the Amiga Made Easy 24

by Francis M. Gardino

A look at dithering options available on the Amiga.

Sculpt Script 26

by Christian Aubert

Creating images using Sculpt-Animate 4-D.

The Art Department 38

by R. Shamms Mortier

ASDG's new color image processing software system.

Scene Generator 40

by R. Shamms Mortier

Brett Casebolt's new and improved Scene Generator V2.10.



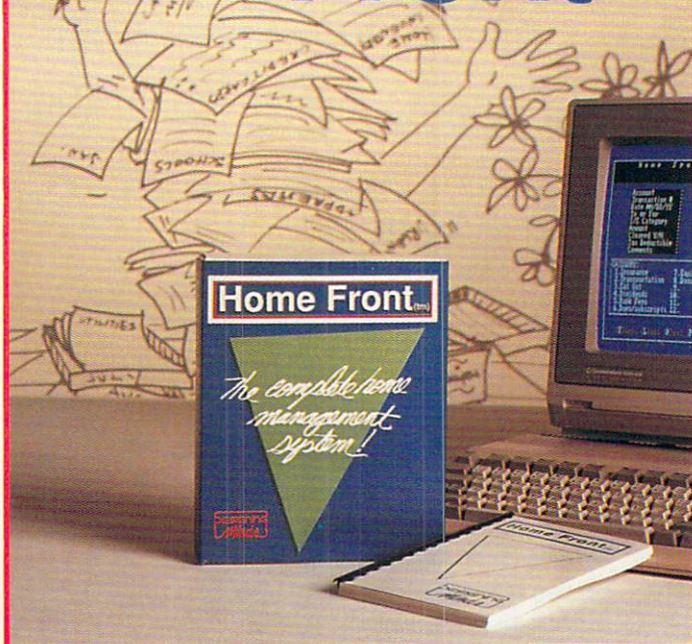
Image created with Scene Generator.

Breaking the Color Limit with PageRender3D 42

by R. Shamms Mortier

No other Amiga animation/3-D rendering package offers such an impressive list of features.

Hometm Front



The Complete Home Management System

Accounting: Advanced Home Accounting System

- Multiple Accounts Checking, Savings, Credit Cards Etc.
- Definable Categories For Incomes And Expenses.
- Easy Carry Over To The Next Fiscal Period.
- Track Tax Deductible Items Making Tax Time Easier.
- More Reports Than Other Amiga Accounting Systems.
- Graphs Income Expense And Budgets.
- Fully Indexed Files, Never Any Waiting!
- Split Transactions
- Periodic Compaction
- Monthly And Yearly Budgets
- Account Reconciliation

Forecasting: Verify Utility And Credit Card Charges. Future Values Of Investments. Print Complete Amortization Tables.

Inventory: Track Vital Detail On Important Household Items. Generate Reports Showing Home & Inventory Worth.

Recipes: Manage, Manipulate And Convert Recipes. Organize Weekly Menus & Print The Shopping List.

Address: Complete Name And Address Manager And More. Remember Any Important Date. Prints Mailing Labels Too!



only \$99⁰⁰

Call Or Write For More Information
Designing Minds, Inc.
3006 North Main, Logan, Utah 84321 (801) 753-4947
No Copy Protection, Installs On Any Hard Disk

Amazing^{AMIGA} COMPUTING

Amazing Computing For The Commodore AMIGATM

ADMINISTRATION

Publisher: Joyce Hicks
Assistant Publisher: Robert J. Hicks
Admin. Assistant: Alisa Hammond
Circulation Manager: Doris Gamble
Asst. Circulation: Brigitte Renee Plante
Corporate Trainer: Virginia Terry Hicks
Traffic Manager: Robert Gamble
International Coordinator: Donna Viveiros
Marketing Manager: Ernest P. Viveiros Sr.
Marketing Associate: Greg Young
Marketing Assistant: Lisa Friedlander
Programming Artist: E. Paul

EDITORIAL

Managing Editor: Don Hicks
Associate Editor: Elizabeth Fedorzyn
Hardware Editor: Ernest P. Viveiros Sr.
Technical Editor: J. Michael Morrison
Technical Associate: Aimée B. Abren
Video Consultant: Frank McMahon
Art Director: William Fries
Photographer: Paul Michael
Illustrator: Brian Fox
Graphic Designer: Kim Kerrigan
Research & Editorial Support: Marilyn Gagne
Production Assistant: Melissa-Mae Viveiros

ADVERTISING SALES

Advertising Manager: Donna Marie

1-508-678-4200
1-800-345-3360
FAX 1-508-675-6002

SPECIAL THANKS TO:

Buddy Terrell & Byrd Press
Bob at Riverside Art, Ltd.
Swansea One Hour Photo
Pride Offset, Warwick, RI
Mach 1 Photo

Amazing Computing For The Commodore AmigaTM (ISSN 0886-9480) is published monthly by PIM Publications, Inc., Carrant Road, P.O. Box 869, Fall River, MA 02722-0869.

Subscriptions in the U.S., 12 issues for \$24.00; in Canada & Mexico surface, \$34.00; foreign surface for \$44.00.

Second-Class Postage paid at Fall River, MA 02722 and additional mailing offices.

POSTMASTER: Send address changes to PIM Publications Inc., P.O. Box 869, Fall River, MA 02722-0869. Printed in the U.S.A. Copyright© August 1990 by PIM Publications, Inc. All rights reserved.

First Class or Air Mail rates available upon request. PIM Publications, Inc. maintains the right to refuse any advertising.

Pim Publications Inc. is not obligated to return unsolicited materials. All requested returns must be received with a Self Addressed Stamped Mailer.

Send article submissions in both manuscript and disk format with your name, address, telephone, and Social Security Number on each to the Associate Editor. Requests for Author's Guides should be directed to the address listed above.

AMIGATM is a registered trademark of Commodore-Amiga, Inc.

EDITORIAL CONTENT

AMIGA GRAPHICS

The Amiga has always been appreciated for its significant graphics capabilities. We can see this in the way other computer companies have worked overtime to provide similar capability (at a higher cost). It is not until we examine the many facets of the Amiga, however, that we appreciate how deeply graphics are involved in every stage of Amiga computing.

Amiga's early inventor, Jay Miner, wanted a computer which could perform some of the advanced graphics and simulation techniques he had seen in extremely expensive flight trainer programs. The professional flight simulator programs were fast and very graphics intensive. By establishing this goal, Mr. Miner's efforts and creativity helped conceive the Amiga. He reached for a single star and opened an entire galaxy for the rest of us to explore.

The Amiga has not only provided a great assortment of flight simulators, from WWII scenarios to those involving space flight, but its inherent graphics capabilities have pushed the Amiga into video, the arts, and more.

BEST-KEPT SECRET

Commodore Business Machines maintains that the Amiga is one of the best-kept secrets in the computer industry. Unfortunately, they are correct. However, Amiga graphics are extremely popular. Millions of people have watched movies such as *Total Recall* and *Three Men And A Baby*. Millions more have seen *Max Headroom*, *Amazing Stories*, or *My Secret Identity* on television. Yet, few of those millions are aware that these productions have made use of Amiga graphics.

During a recent discussion with an Amiga developer in the Hollywood area, I asked if we could get a few stories on what people are doing with the Amiga. He said no one is talking. Although the Amiga is being used quite heavily in a lot of the pre-production and post-production houses, few want the information known. It appears that, in many cases, the Amiga is taking the place of some very expensive

equipment. The artists are reluctant to let anyone know, as they continue to invoice their customers at the more traditional and expensive rates. Perhaps the Amiga is performing its task too well.

There are artists, developers, and organizations which are discovering the unique nature of the Amiga daily. These people have transformed what is promised in the specifications for the Amiga into the "living", articulate drawings, renderings, and animations we see today.

PORTRAIT OF AN ARTIST

When discussing Amiga graphics, it is impossible not to mention the name Jim Sachs. Mr. Sachs is the most famous computer artist in the Amiga community. He first taught himself graphics and machine language on a Commodore 64, and is a pioneer in Amiga art. A former Air Force pilot with no formal art training, Jim Sachs created many of the great drawings and renderings of the early days of the Amiga.

Mr. Sachs' accomplishments include the farsighted creativity displayed in the CinemaWare hit *Defender Of The Crown*. His early work was found on bulletin board systems and magazine covers. He was a constant visitor to Amiga user group meetings, and an inspiration to every other artist working with the Amiga. He created the Mazda pictures used by Aegis, *Sachs Castle 1990* (his then hoped-for expansion to his mountain home), several views of Porsches and motorcycles, and even a few scenes for a prospective new game—*Time Crystal*.

In an interview that appeared in *Amazing Computing's* April, 1987 issue, Jim Sachs commented, "It turns out that the kind of art that I'm good at—and the kind of art I like to do—translates very well to the computer screen."

"Detail, contrast, mostly contrast... On paper, you start with something that is white and put down dark colors. With the computer screen it's just the opposite; you start out with a black screen and paint with points of light."

Mr. Sachs has not kept his unique interpretation of graphics and its relation-

ship to computers a secret. He is a major teacher at assorted AmiEXPOs as well as a constant seminar instructor for universities. Those interested in exploring the depths of art as it appears on our Amiga screens should not miss the opportunity to see and hear him.

HELP WANTED:

ARTISTS AND ANIMATORS

I often receive phone calls from developers looking for a particular talent or individual to help them complete a project. I was especially happy when Alexander Eckelberry called from Oxxi inc. with a request for any artist who would be willing to help him test a new product. It appears that the people at Oxxi are seriously updating one of the products that they acquired, and would like to see diverse new graphics created with it for its introduction.

Although I do not have access to the new product myself, I can assure you that it is always a thrill to work on a new project before its introduction. Not only are you on the front edge of the Amiga market, but you have the chance to make suggestions and institute changes that will make the program easier to use.

Serious artists who feel their work will benefit this new product's introduction should send their art on disk, as well as a brief description of their efforts to:

Oxxi inc.
1339 East 28th Street
Long Beach, CA 90806

ARTIST SERVICES

Computimation, a product of Curt Kass' fertile imagination, is a collaboration of artists for hire. Mr. Kass has combined the talents of animators, still artists, and musicians to create a group of artist available for large or small graphics needs. Working with computer-based systems, these artists are attempting to make it simple for advertisers and businesses to use high-quality Amiga graphics and animations in their presentations and advertising.

(continued on page 80)

The Best Assembler Macro68

Suggested retail price: US\$150

Macro68 is a powerful new assembler for the entire line of Amiga personal computers.

Macro68 supports the entire Motorola M68000 Family including the MC68030, MC68882 FPU, and MC68851 MMU. The Amiga Copper is supported also.

This fast, multi-pass assembler uses the new Motorola M68000 Family assembly language syntax, and comes with a utility to convert old-style syntax source code painlessly. The new syntax was developed by Motorola specifically to support the addressing capabilities of the new generation of CPUs.

Macro68 boasts macro power unparalleled in products of this class. There are many new and innovative assembler directives. For instance, a special structure offset directive assures maximum compatibility with the Amiga's interface conventions. A user-accessible file provides the ability to customize directives and run-time messages from the assembler. An AREXX(tm) interface provides "real-time" communication with the editor of your choice. A number of directives enable **Macro68** to communicate with AmigaDos(tm).

Possibly the most unique feature of **Macro68** is the use of a shared-library, which allows *resident* preassembled include files for incredibly fast assemblies.

Macro68 is compatible with the directives used by most popular assemblers. Output file formats include executable object, linkable object, binary image, and Motorola S records.

Requires at least
1 meg of memory.

ReSource,
the powerful
disassembler for
the Amiga that has
received rave reviews,
now has a big brother.

Like the original version,
ReSource'030 will tear apart
your code like no other program.
And it will do so even faster now,
because **ReSource'030** is written in
native MC68030 code. This means that
it won't run on a vanilla 68000, but will fly
on an A3000, or another machine with a
68020/030 board.

ReSource'030 supports the new Motorola M68000
Family assembly language syntax, and is a perfect
companion to **Macro68**.

If you're new to **ReSource**, here are a few facts:
ReSource is an intelligent interactive disassembler for the Amiga
programmer. **ReSource** will enable you to explore the Amiga. Find
out how your favorite program works. Examine your own compiled code.

ReSource will load/save *any* file, read disk tracks, or disassemble directly
from memory. Symbols are created automatically, and virtually *all* Amiga symbol
bases are supported. Additionally, you may create your own symbol bases.

"If you're serious about disassembling code, look no further!"

The original **ReSource** continues to be available for owners of 68000 based machines.
Both versions of **ReSource** require at least 1 meg of ram.
Suggested retail prices: Original **ReSource**, US\$95, **ReSource'030**, US\$150

ReSource The Best Disassembler



The Puzzle Factory, Inc.

P.O. Box 986

Veneta, OR 97487

Orders: (800) 828-9952

Customer Service: (503) 935-3709

Distributors for the U.S. and Canada **Dealer Inquires Invited**

"Quality software tools for the Amiga"



VISA, MasterCard, check or money order accepted - no CODs.

Amiga and AmigaDOS are trademarks of Commodore-Amiga, Inc.



Feedback

CENSUS REPORTS

As a response to your editorial in the June, 1990 issue of AC, I would like to submit the following.

I started building electronics projects back in high school and have been tinkering ever since. My B.S.E.E. from Purdue has helped me greatly in my quest to build bigger and better projects. I have tried to find inexpensive solutions to some of computer/electronics hobbyist's current problems. I guess I've always considered myself a backyard pioneer so when I bought my first Amiga back in 1987, it fit right in with the greater scheme of things.

In the last couple of years, I have been concentrating on building projects for my Amiga. These have ranged from such mundane things as a switch box for the selection of the mouse to an amplifier for sound output. I like to get the most out of my designs while still keeping them compact and efficient. So, my projects do go a little further than just 'bare bones' solutions.

The mouse switch box I designed will allow the mouse port to connect to any of three devices, two of which can also be used with the joystick port. The stereo amplifier has an input multiplexer to allow sound input from an Amiga, a 'Walkman'-type portable radio, or a television. This makes working with my Amiga pleasant when it's not being used to blast the output of the latest fast-action arcade games. The amplifier also includes a preamp to allow adjustment of the volume, balance, treble, bass, loudness, and separation of the selected input signal. I also designed a MIDI interface. It doesn't just have an input and output, it has one input, three individual buffered outputs, and a switchable out/thru connection all running at about 15 milliamps total.

I purchased an Amiga 2000 on April 9th. Is this "... within ninety days of the AmigaVision announcement ..." as men-

tioned in the "World of Amiga..." article (v5.6, pg. 47)? If it is, what do I need to do to receive a copy of AmigaVision?

Jim Lundy
Frisco, TX

—AmigaVision was announced on April 24, 1990, but wasn't released until June 30, 1990. The announcement applied only to those who purchased an Amiga between the time of the announcement and the actual release of AmigaVision. Refer to Amazing Computing's August issue (p. 94) for more details.—ED

SPEEDING UP CBM'S OSCILLATOR

Most magazines have reviewed all the 68030 boards, showing Commodore's are slower than the others. Well here's a little tip to get more speed out of the A2630 Card... CHANGE THE CRYSTAL.

It is possible to replace the 25 MHz oscillator with a faster version. The only problem is locating the part. I tried three electronics suppliers, and found only oscillators of less than 24, or more than 30 MHz, and was starting to give up hope. While installing some boards in my A2500, I found that there was a 28.63-MHz oscillator on the motherboard. I was able to obtain the oscillator from a Commodore service shop.

Installation is quite simple: 1) remove your 68030 Board, 2) desolder & remove the 4-pin, 25-MHz crystal on the card (using a proper desoldering tool), 3) resolder 28.63-MHz crystal in its place, and 4) replace the card. Test the unit fully.

Just for the record, I have the A2630 card from Commodore w/4 megs, 6.2 motherboard, A2091A HD controller w/2 megs, A2052 2MB RAM card, and Mimetics' FrameBuffer.

I have installed a 28.63 MHz oscillator, replacing the 25 MHz version shipped with

the A2630. I haven't had difficulties on my machine at 28 MHz. Ray-tracing packages reveal a notable difference between a stock A2630 and the modified 28 MHz. I would say that you should notice approximately a 10% speed increase.

I have also been using SetCpu with the options Burst and Cache on. When using the Ronin CPU/Memtest, I get 11.50-11.91 reading, which beats the stock reading of 10.5-10.96. This places the A2630 card at about the same level as the GVP28, although I think Commodore's might still be a tad slower with the 100 ns chips.

SetCpu V1.6 is a PD program (available on most BBS's) by Dave Haynie, and is a must for any accelerator card owner.

I have done 3 installations of a 28.63 MHz oscillator. On one board without the shielding (the first few A2630 Boards were shipped without the metal case around the main chips), the machine would crash/freeze after a 3-5 minute period. I don't know if it was the '030 Board, or the older A2090 HD controller that caused the Gurus, but I replaced the 25 MHz on the main processor and moved the 28.63 MHz crystal to the math coprocessor (and moved the jumper). It has boosted the speed of Sculpt-Animate 4D, over a stock A2630 card.

For the price of the oscillator (roughly \$7.00-\$12.00), I feel this is a worthwhile modification to perform, and only requires a small amount of time to try. Please note that this modification will void your warranty unless you have it done at a Commodore service center.

Andrew H. Hochheimer
Wallaceburg, Ontario

DPRINT II VERSUS DPI

Unfortunately Deluxe Print II does not handle printers quite as well as we all would like it to. A call to Electronic Arts reveals that DPrint II expects a 120 x 120

GVP Announces a Technological Breakthrough...

SERIES II™ THE NEXT GENERATION

in SCSI and RAM Controllers for the A2000®

IMPACT

Series II

GVP's New **SERIES II A2000 SCSI and RAM Expansion Controllers** provide the ultimate hard disk and RAM expansion solution for the A2000. Choose from two new models:

The Series II A2000 SCSI "Hard-Disk + RAM-Card"

- State-of-the-Art integration packs a high performance SCSI controller, 8MB FAST RAM Expansion and a 3.5" hard disk drive INTO A SINGLE A2000 EXPANSION SLOT!! Saves BOTH a valuable expansion slot and a peripheral bay!
- Incredible SCSI hard disk performance achieved through GVP's innovative new custom chip design, which provides DMA performance and unique direct dual port memory access to FAST RAM, eliminating typical DMA side effects under heavy graphics load.
- Easy-to-install SIMM memory modules allow flexible memory configurations from ZERO through 8MB. Supports 6MB FAST RAM configuration for BridgeBoard users.
- NEW **FAAASTROM™** SCSI Driver offers optimum performance and includes such features as:
 - ✓ Supports virtually any SCSI device including, CD-ROMs, Tape Drives, IOMEGA Bernoulli drives, etc.
 - ✓ Fully implements SCSI Disconnect/Reconnect protocol, allowing overlapping SCSI commands to be executed.

- ✓ Fully implements Commodore's Rigid Disk Block (RDB) standard as well as the new DIRECT SCSI interface standard.
- ✓ Removable media drive support. Automatically senses cartridge changes and informs AmigaDOS, ensuring safe and reliable use of removable media SCSI drives.
- ✓ Allows Direct AUTOBOOT from Fast File System Partition.
- New **INTUITION COMPATIBLE SCSI** installation and "tuning" utility included. Major features include:
 - ✓ ICON and gadget based INTUITION interface.
 - ✓ Bad Block Remapping of hard drives.
 - ✓ Auto or manual hard drive partitioning and AmigaDOS formatting.
 - ✓ Read and modify existing RDB parameters on hard disk.
 - ✓ Simplest and Easiest SCSI installation in the industry.
- Low parts count (through VLSI Integration) EQUALS: lower power, higher reliability, longer life and ultimate PRICE/PERFORMANCE! See TRADE-UP offer.

The Series II A2000 SCSI "Hard-Disk-Card"

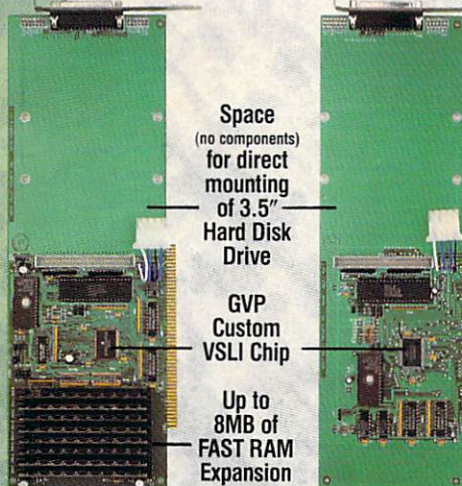
- Same as above but without the 8MB FAST RAM capability.
- Specially designed for those users who don't need memory expansion but still need maximum hard disk performance at a budget price.
- **UNBEATABLE VALUE.** See \$99 trade-up offer!

GVP's New **FAAASTROM** SCSI driver and installation software is also available as an upgrade kit for GVP's original IMPACT SCSI controllers, for ONLY \$49.95. Offers major performance increase over previous GVP AUTOBOOT EPROMs.

New Series II 48MB Removable media hard disk drive. GVP now also offers the NEXT GENERATION removable media hard disk drive which offers increased capacity (48MB formatted) and major technological advances in cartridge air flow filtering design and robustness. Call for details.

Hard-Disk+RAM-Card

Hard-Disk-Card



SCSI TIMES

1990

The ULTIMATE Trade-Up Offer???

GVP today introduced its new Series II product line and announced a bold new trade-up program, which is certain to further bolster GVP's dominant market share in the Amiga hard drive market.

Details of GVP's new TRADE-UP program are as follows:

- For \$109 plus \$6 for shipping (USA only), end-users can trade-up to the new GVP Series II SCSI "Hard-Disk-Card" (without drive) by simply sending in their present SCSI controller (from ANY manufacturer) together with a money order or certified check, payable directly to GVP.
- All trade-in controllers must be sent to GVP **FREIGHT PREPAID.**
- Owners of any GVP or Commodore SCSI controllers, are eligible for an additional \$10 rebate, if they trade-in a **FULLY FUNCTIONAL** and working controller. These owners need to submit a certified check/money order for \$105 only.
- For an additional \$39, existing controllers can be traded-up to the new Series II "Hard-Disk + RAM-Card," which includes the 8MB FAST RAM expansion capability, populated with ZERO RAM.

"Let's Standardize"

GVP

GREAT VALLEY PRODUCTS INC.

600 Clark Avenue, King of Prussia, PA 19406

For more information, or for nearest dealer, call today. Dealer inquiries welcome.

Tel. (215) 337-8770 • FAX (215) 337-9922

Series II, FAAASTROM and GVP are trademarks of Great Valley Products, Inc. Amiga and A2000 are registered trademarks of Commodore-Amiga, Inc.

Circle 123 on Reader Service card.

Hypertext

for AMIGA

Version 2.1: Mixed text styles!
Images in documents! Colors!
Enhanced Interface! CLI access!

"Will certainly whet a lot of
HyperAppetites"

Neil Randall,
Amigaworld 1/90

"Its flexibility far
exceeds any other
program that I've used
on any computer."

Robert Klimaszewski,
Amazing V5.1



THINKER

Write, design, plan. Multimedia Idea
Processor with HyperText! ARexx

Upgrades

Version 2.1

1.X -> 2.1 \$25

2.0 -> 2.1 \$10

\$80

Poor Person Software

3721 Starr King Circle, Dept 5

Palo Alto, CA 94306

(415)-493-7234

Circle 129 on Reader Service card.

DPI printer to be attached. Therefore, anything higher than 120 DPI (dots per inch) is classified as hi-res (news to me). The workaround for this is to go into Preferences for the printer and set the density to a setting low enough to match 120 DPI. If the printer driver does not have any settings such as this then you will need to figure out what combination of settings will achieve the desired results. The control is best understood in terms of width and height since each interaction functions differently.

WIDTH: This is the easiest part to control. Amiga preferences can be manipulated to print this at full width simply by setting the Graphics 2 limits to PIXEL and setting the width limits according to the following equation.

$$\text{Width_limit_in_Pixels} = \text{Desired_width_in_inches} * \text{Printer_Width_DPI}$$

For example the HP PaintJet is 180 x 180 DPI. Therefore, to print 8.5 inches wide on the HP PaintJet, the pixel width limit should be set to 1530.

Ex. 1530 pixels = 8.5 inches * 180 DPI
Next, when you actually attempt to print the document increase the width until the last setting before it tells you that you can't print to the dimensions specified. On the HP PaintJet this is 5.3 inches.

HEIGHT: This one is difficult to control. The easiest way to explain it is that you will be adjusting the Height limit in Amiga Preferences so that one inch on the hi-res printer will be equal to one inch on a low-res printer.

Essentially what happens here is that each printing pass that DPrint II makes will be forced to the dimensions specified in Amiga preferences. Therefore you will be making this forced dimension equal to that of one pass of a low-res printer. On the HP PaintJet this magic number was eight pixels.

Once you have found the correct magic number for Amiga preferences your height will be controlled by the height requester in DPrint II. Also to make sure that there is no additional interaction you should turn off the Proportional gadget in DPrint II.

In summary here is how I print a 8.5 x 11 sign on the HP PaintJet.

- 1) go into printer preferences.
- 2) go into Graphics 2 screen.
- 3) click on the limits gadget labeled PIXEL.
- 4) click on the width limit requester and enter 1530.
- 5) click on the height limit requester and enter 8.
- 6) click on smoothing on gadget (personnel preference).
- 7) click on OK (to exit graphic 2 screen).
- 8) click on OK (to exit change printer screen).
- 9) click on use (to exit Preferences using current settings without saving).
- 10) enter DPrint II and load desired file.
- 11) select print.
- 12) enter 5.3 for width requester (max. width without complaints).
- 13) enter 11 for height requester (actual dimension once magic number was found).
- 14) click on proportional to turn off.
- 15) click on OK to print.

Thank you,
Ron Black
Laurinburg, NC

A FIX FOR PAGESTREAM FONTS:

This is in response to John Steiner's article "PageStream 1.8" in the June issue. The solution to his assignment problem of PageStreamFonts: can be found not in the Set/Save Paths option, but in the Font Manager, even though you no longer need to make that assignment. Since I have not had any problems with the Set/Save Path option, it was not necessary for me to make an assignment for PageStreamDrivers.

My second point relates to his comment that "sizing and cropping of the EPS images is impossible". While cropping an EPS image may be impossible, I have a technique which makes sizing relatively easy. This technique assumes that you are working with the original artwork in Professional Draw. After you have completed your illustration move it to the upper left-hand corner of the page. Then select Alter Current Page and resize the page so that it is just big enough to hold your illustration. It is also a good idea to keep the size of the image relatively small, about 2" x 2", so that when it is brought into PageStream the box fits easily onto the page. Next you have to print the EPS file. Make sure to print only the current page. If you print the entire folio, even if it is only one page long, PageStream will offset the position of your image on the printed page. Finally, when you import the EPS file into PageStream you will be given a box representing the exact size of your illustration. At this point resizing your image should not require any guesswork.

Sincerely,
Michael Rubino
E. Northport, NY

•AC•

All letters are subject to editing. Questions or comments should be sent to:

Amazing Computing
c/o PIM Publications
P.O. Box 869
Fall River, MA 02722-9970
Attn: FEEDBACK

Readers whose letters are published will receive five public domain disks **FREE**.

Color Scanning This Good Has Never Been Easier Or Less Expensive

Color scanning on the Amiga personal computer has never been so good, easy or inexpensive. Sharp's new JX-100 portable color scanner (shown in Figure 1) is a price/quality breakthrough and ASDG's ScanLab/100 (the official JX-100 controller for the Amiga) puts it to work on your Amiga 500, 1000*, 2000 or 2500.

Getting the image you want is easy with the JX-100. Just place your original under the scanner and zero in on what you want using ScanLab/100's preview mode (Figure 2). Then, in fine mode (Figure 3), you can scan the area you selected at up to 200 dots per inch in 64 shades of gray or 262,144 colors. Figures 1 and 4 shows unretouched results, both being 18-bitplane images.



Figure 1.

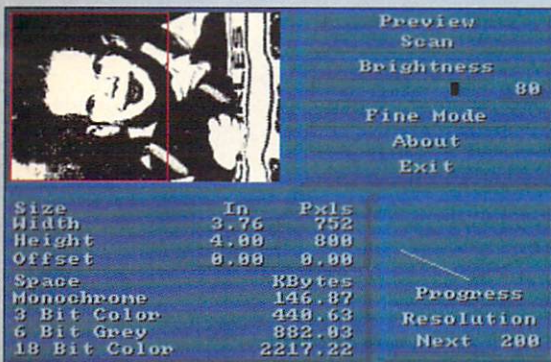


Figure 2.

ScanLab/100 is the best video rendering tool ever produced for the

Amiga. It supports more video modes (including 4096 colors in Hi-Res) than any other Amiga program as well as a host of other unique features. These include:

- 208 Video Modes. ● PAL and NTSC in the same program!
- Six dithering types. ● Full color balancing including variable Gamma correction.
- Loads any IFF image (including SHAM, Dynamic Hi-Res and Dynamic HAM) up to 24-bitplanes deep and 1280 by 1280 pixels in

size. ● New video modes including A-RES (4096 colors in hi-res) and A-HAM modes. ● Variable digital image reduction. ● Overscan screens up to 768 by 592. ● 68020 and 68030 compatible.

Scan Controls		Commands	
18 Bit Color		Scan	
Thresholds		Save	
Resolution	288	Load	
Width	752	Display	
Height	888	Preview	
Space	2217.22	Exit	
Color Controls		Screen Controls	
Balancing		Low Res/Overscan	
Dither: Sierra (3)		NTSC/OverScan	
Image Controls		Colors	
Portrait	Reduce	HAM	
Image Size		Execute	
W: 768	H: 488		

Figure 3.



Figure 4.

Best of all, in the United States, ScanLab/100 is included in the \$995 list price of the Sharp JX-100 scanner. For desktop publishing, presentations or desktop video, the Sharp JX-100 and ScanLab/100 is the best color input system in its price range! Call now.**

ASDG Incorporated 925 Stewart Street Madison, WI 53713 Voice (608) 273 - 6585

ScanLab/100 is a registered trademark of ASDG Incorporated. Amiga is a registered trademark of Commodore-Amiga, Inc. This ad was created using ASDG's Art Department, ScanLab/100, and ReSEP along with Gold Disk's Professional Page. *Requires gender changer on the A1000. **Dealer inquiries welcome.

Circle 108 on Reader Service card.

Keyboard Controlled Sequencer 3.0

by Phil Saunders

DR. T'S KEYBOARD CONTROLLED SEQUENCER 3.0 INCORPORATES A NUMBER OF significant changes from KCS 1.6. Some of these are visual, like the new pull-down menus and remote control section on the Track Play screen. Some add significant new functions, like the implementation of the Multi-Program Environment (MPE) and increased timing resolution of the clock. But the most notable improvement is in the overall feel of the program: KCS 3.0 now runs like a finely tuned race car.

The quirks and rough edges that were the legacy of KCS's Commodore 64 and Atari ST heritage are for the most part gone. KCS 3.0 now feels like an Amiga program, while retaining full compatibility with the Atari ST version. The addition of the MPE allows Dr. T's programs to work together seamlessly as part of one powerful, modular MIDI system.

The screen layout and user interface of KCS 3.0 are greatly improved. The confusing arrows which were used to scroll through the event list in V1.6 have been replaced by a standard Amiga scroll bar. The mouse is much more responsive when used to highlight a section of the event list. Many of the commands which were on-screen buttons now appear in well-organ-

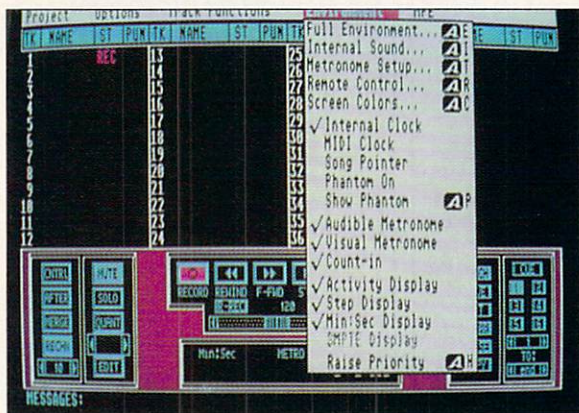
ized pull-down menus. Keyboard equivalents are also provided for many commands. The layout of the items in menus conforms to Amiga standards, with file options grouped under the Project menu. KCS 3.0 reads and writes MIDI files directly, instead of through a conversion program, as in version 1.6. Unfortunately, there is still no provision for importing or exporting SMUS files.

The remote control panel in Track mode is another useful new feature. It provides a set of transport and punch in/out controls which can be detached from the Track Play screen and slid down over another screen. This is especially useful when multitasking with a patch editor or other MIDI program, because the transport

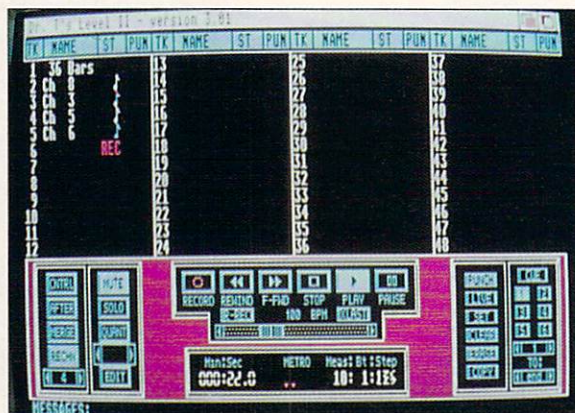
controls allow you to play back and record while editing sounds and auditioning patches. The remote control also includes "erase last track" and "rerecord" buttons, perfect for ham-fisted keyboardists like myself.

One quibble of mine is that, while the erase track menu option can be selected by the mouse, the program then requires you to type "Y" or "N" to confirm the erasure. This prevents accidental erasure, but requires you to be close to the computer keyboard at all times. The "MIDI-keys" function permits footswitches connected to a synthesizer to control KCS's transport controls and rerecord features. I bought two footswitches for my DX-7, and can now rerecord a track by just stepping on a switch. This is an incredibly convenient feature which already existed in KCS 1.6.

KCS has always been one of the most powerful (and bug-free) Amiga MIDI sequencers. One of its biggest advantages is that it doesn't force the user to structure his or her music in any particular way. If you like the multitrack tape recorder metaphor, the Track mode provides 48 tracks that run the length of the entire song. If you create



KCS 3.0 features pull-down menus.



The Track Play screen has been redesigned.

Incorporating an improved user interface and a number of new features, KCS 3.0 now runs like a finely tuned race car.

music by stringing together patterns, the Open mode provides a flexible way of doing this. The Song mode also lets the user combine Track and Open mode sequences. These three distinct modes of operation make KCS tremendously flexible.

One feature KCS 3.0 doesn't have is graphic editing, which most of its competitors provide in one form or another. Editing is performed in KCS by editing numbers in an event list. While this is not always as convenient as graphic editing, KCS provides more ways to manipulate the MIDI data. The KCS approach is more powerful than most graphic editors because you can address all the data by using an algorithm. With KCS (and particularly with KCS Level II), it is a simple matter to change all the C's in a sequence to C#s, while at the same time increasing their velocity and shifting them forward in time. Graphic sequence editing simply can't provide that kind of detailed editing as easily. All in all, KCS has more editing power than any other Amiga sequencer.

If graphic editing is important to you, Dr. T's MPE provides a solution. TIGER is a separate graphic editor that can be integrated with KCS 3.0. It provides real-time graphic editing using "piano roll" notation. TIGER can edit any standard format MIDI file, but is particularly well suited for KCS. TIGER can add and delete notes while a KCS sequence is playing and can also edit pitch bend and controller data. The MPE also allows TIGER and KCS to share the same data storage area, so changes made in one program are automatically incorporated into the other. Using KCS 3.0 with TIGER is intended to provide full graphic editing along with KCS's powerful event list editing. TIGER should be available for the Amiga by fall.

What is the Multi-Program Environment? Essentially, it is a way to allow

multitasking MIDI programs to access the same data in real time. Data can be routed from one program to another, modified, and then recorded into KCS and output from the MIDI interface. The MPE allows programs to edit the same data, so it is not necessary to import and export files from one program to the next. The MPE lets KCS 3.0 interact with other Dr. T's programs like The Copyist, TIGER, Phantom (a SMPTE-synchronization), AutoMix (available with KCS & Level II), and patch editors. Essentially, the MPE turns all compatible programs into modules in an integrated MIDI system. All the programs work together on a common set of data. The result is a kind of "super-sequencer" that can add graphic editing, score printing, patch editing, and SMPTE capabilities to the basic KCS functions. The user can choose programs to fit his or her particular needs with the confidence that all the programs will work together.

I have some reservations about the MPE. On the one hand, it promises to be the most capable MIDI system available for the Amiga. If all the parts work as planned, KCS 3.0 will be the centerpiece of a powerful, integrated environment which takes full advantage of the Amiga's multitasking abilities. No other Amiga sequencer has as much potential. But there are also drawbacks to the approach Dr. T's has taken. While other manufacturers can write MPE-compatible programs, it is unlikely that many will be willing to conform to another company's specifications. This problem is evident on the Atari ST, which has a number of competing multitasking systems. Dr. T's has jumped out in front with a capable system, most of which is available now. It is an open question whether other companies will adapt their programs to the MPE or whether they will wait for the official Commodore MIDI manager. [Ed. note: A spokes-

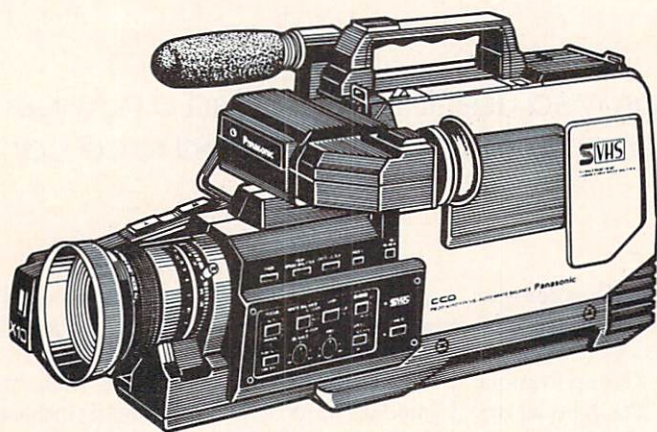
person for Dr. T's has stated that the MPE was in the works before any mention of Commodore's MIDI manager surfaced.]

There is another problem with the modular approach. KCS 3.0 doesn't include SMPTE synchronization or MIDI time code, features available on other professional Amiga sequencers. Instead, the user can add SMPTE synchronization with Phantom, which combines an MPE software module and an Amiga MIDI interface/SMPTE synchronizer. Phantom uses a direct SMPTE lockup to synchronize with the Amiga, instead of using MIDI Time Code like other Amiga sequencers. A Dr. T's representative said that the Phantom approach resulted in quicker and more accurate synchronization because synchronization signals were not competing with other MIDI data. But since Phantom uses both hardware and software, it may not be compatible with other MIDI sequencers. A second, different SMPTE interface will be necessary to sync Music-X or Passport to SMPTE. This is a big drawback for musicians who routinely use more than one sequencer. Of course, other MIDI programs would still work with Phantom's MIDI interface and could interchange files through the standard MIDI file format. Dr. T's reports that Gold Disk, Elan, and Steinberg are preparing programs that will be compatible with the Phantom interface.

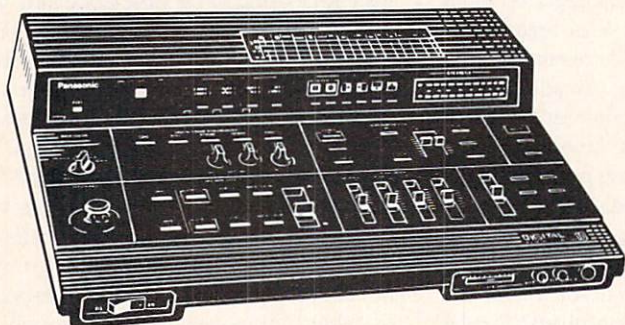
There are also the questions of integration and expense. While a modular system is very flexible, the parts don't always fit together as nicely as they do in an all-in-one sequencer. The cost of buying additional modules for SMPTE and for graphic editing can add up quickly. Programs that are not MPE-compatible will still multitask with KCS if they follow Amiga rules and are well-behaved. Dr. T's has released updates for earlier patch editors so they will be fully MPE-compatible. Updates for patch edi-



Video For Amiga



Panasonic *AV*
Professional/Industrial Video



G & G Technologies is one of the largest Panasonic Industrial video dealers in the country. We can provide you with all your video needs, whether you are a weekend wedding photographer or a broadcast pro.

We are strong advocates of the S-VHS format and Panasonic Industrial is the industry leader for those products. If you need an S-VHS camcorder, or an S-VHS edit system, call us and we will explain the various options. If you are currently working in another format, we can show you how to adapt your gear to the S-VHS format. The high resolution of S-VHS is ideal for Amiga applications. Panasonic's VJ-MX12 mixer and an Amiga can make for some terrific special effects to enhance your productions.

Call (or stop in) and let us help you with all your video needs!

G&G Technologies

800-422-2920

333 North Street, Teterboro, New Jersey 07608

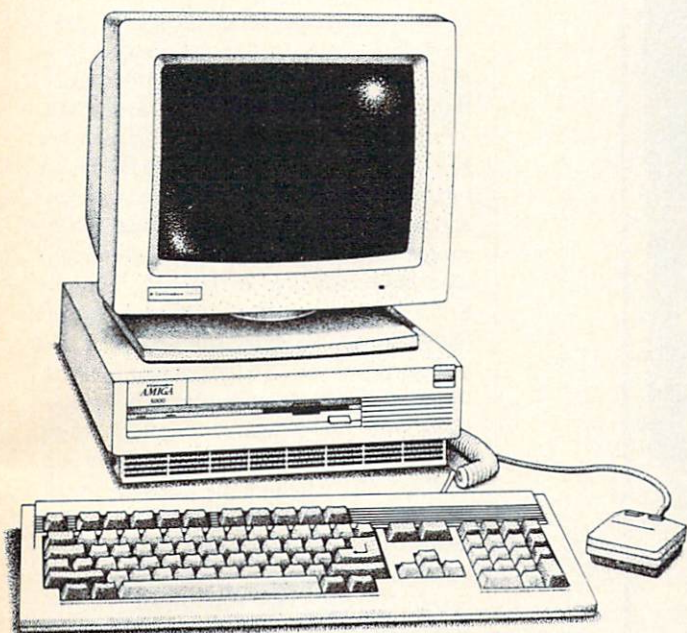
Store hours are Monday-Friday, 9am-5:30pm (EST)

We accept MasterCard, Visa, American Express & COD.

In New Jersey please call 201-288-8900. Our fax number is 201-288-6881.

Be sure to visit our booth at Video Expo in New York, September 11-13.

Amiga For Video



Commodore®
AMIGA®
3000

We specialize in Amiga packages for video applications. If you have been trying to find an authorized Amiga dealership that knows video products and how they relate to the Amiga, then call us! Our knowledgeable sales staff will assist you in creating a video package that will suit your needs AND your budget.

We carry the complete Commodore Amiga line, a full selection of genlocks for the Amiga, and all the video software for any type of video project. If you are in the area, we invite you to stop in and see the new Amiga 3000 demonstrated in our state-of-the-art showroom.

G&G Technologies

800-422-2920

333 North Street, Teterboro, New Jersey 07608

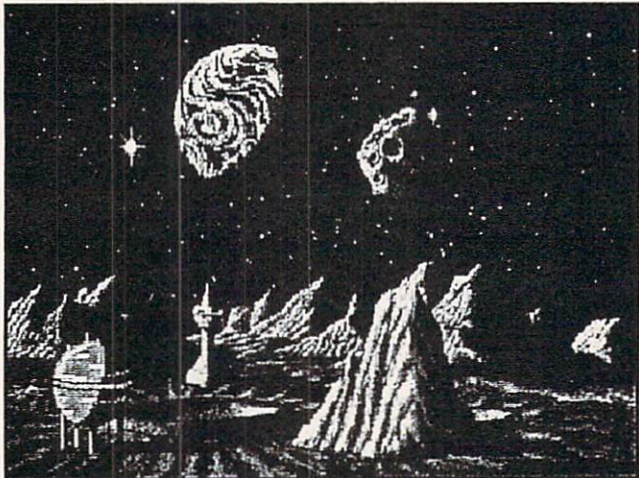
Store hours are Monday-Friday, 9am-5:30pm (EST)

We accept MasterCard, Visa, American Express & COD.

In New Jersey please call 201-288-8900. Our fax number is 201-288-6881.

Be sure to visit our booth at Video Expo in New York, September 11-13.

Silver Fox Software Presents:
LUNAR Construction Disks



Create your own fantastic scenes of lunar landscapes, tumbling asteroids, and sparkling stars on the Amiga with these high quality, full color images. This two disk set contains over 200 pictures, brushes, and anim brushes -- your only limitation will be your imagination.

To order your copy of LUNAR Construction Disk please send a check or money order for \$25.00 (+\$2.00 P&H) to:

Silver Fox Software
 P.O.Box 551413
 Dallas, Tx. 75355-1413

Call (214) 349-1681 for information and dealer inquiries.

Circle 105 on Reader Service card.

tors are available from Dr. T's by sending the serial number of your disk along with a check for \$25 per program. MPE-compatible versions (V1.63) of The Copyist and Copyist Apprentice are also available for the same upgrade fee. Dr. T's is working with Commodore on the design of a MIDI manager, and it may be that Phantom and KCS will be compatible with the official Commodore system. Despite the questions about compatibility, it is clear that the MPE and KCS 3.0 are a significant step forward for Amiga musicians.

A bonus program included with KCS 3.0 serves as a quick demonstration of the MPE. AutoMix is an MPE-compatible program that allows the user to create automated mixdowns using MIDI volume, pan, or other continuous controller messages. The program has sliders to control volume and pan settings on all 16 MIDI channels. By moving the sliders, you can adjust the settings on synthesizers and record your changes into KCS. KCS will then replay your fader changes along with the song! The program can also save a snapshot of

the current settings. I find AutoMix to be very useful for adjusting the channel volume settings on my Roland MT-32. It would also be useful for controlling any MIDI sound modules. MPE-compatible patch editors will work in a similar way, allowing patch changes and controller adjustments to be recorded directly into KCS sequences.

There isn't space to list all the little changes and improvements programmer David Silver has made in KCS version 3.0. Additional significant changes include improved timing resolution (to 240 parts per quarter note), improved merging/rechannelization capabilities, and provisions for MIDI control of the Fostex R-8 tape recorder. The KCS event list now displays notes in measure/beat/step format, which is much easier to edit. Also worth noting is the manual, which has been completely revised and greatly improved. It includes numerous Amiga screen shots and a full index. One change is the omission of the help screens, which were left out of this version due to memory considerations. Personally, I don't miss them. The well-

written manual more than compensates.

No program is perfect. KCS only supports 16 MIDI channels (other programs use more than one serial port to support 32 or more channels). And while it is the most powerful Amiga sequencer, some programs for the IBM and Macintosh include more editing features. Another caveat: KCS 3.0, like all of Dr. T's Amiga programs, uses key disk copy protection. This can be an important issue, especially if you run several programs using the MPE. Several other Amiga sequencers are not copy-protected. One important note is that KCS 3.0, like KCS 1.6, is remarkably stable. I've experienced no crashes while running it. Dr. T's also reports that KCS 3.0 is compatible with Workbench 2.0, though it may be necessary to turn off the instruction cache when running the program on a 68030. They are working on a patch for the problem, and it may already be fixed by the time this review appears. KCS offers on-line help through the Berklee Macintosh User's Group. If you call their BBS at 617-739-2366 and leave the registration number of your program, Dr. T's will validate you. This gives you access to program updates and sequence files as well as on-line support from the Dr. T's staff.

When I first reviewed KCS 1.6 (in the March '90 issue of AC), I praised its power and lamented its user interface. I felt that it was a powerful program that required too much effort from its users. The new revisions and the addition of the MPE address many of my concerns. Version 3.0 incorporates an improved user interface and a number of new features. The MPE allows TIGER and Phantom to supply the graphic editing and SMPTE features that KCS itself lacks. If you combine KCS 3.0 with other Dr. T products, it is possible to build an extremely powerful, professional MIDI system. Other Amiga sequencers may be flashier and slightly easier to use. But KCS 3.0 gives you the most control over how your music sounds. I recommend it to any Amiga user who needs a professional sequencer.

•AC•

Keyboard Controlled Sequencer v3.0
Dr. T's Music Software, Inc.
 220 Boylston St., #206
 Chestnut Hill, MA 02167
 (617) 244-6954
 Price: \$275.00
 Inquiry #215

NEW PRODUCTS & other neat stuff

compiled by Elizabeth G. Fedorzyn

WHAT'S IN A NAME?

While our own Bandito may have erroneously reported otherwise ("Roomers", v5.8, p. 62), **SAS Institute, Inc.** will hardly be abandoning the Amiga Lattice C Compiler. The parent company, which has always been responsible for the development of the Lattice C Compiler for the Amiga, assumed all product responsibility for the compiler as of July 1, 1990. And, in fact, version 5.10 of the Amiga Lattice C Compiler has just been released.

Now known as the **SAS/C Compiler for AmigaDOS**, version 5.10 will feature, among other enhancements, LSE ARexx support, AmigaDOS 2.0 support, improved user interface, and improved Workbench utilization. Changes have also been made to LC2 to improve generated

code. The speed of LC1 has been improved, especially when generating debugging information with the -d option. BLINK's speed has been improved as well. The new version is available to existing 5.0 customers for an upgrade fee of \$40.00.

SAS Institute is providing full technical support for this recent version only. Also in regards to tech support, effective September 1, 1990 SAS will offer technical support through a 900 number (1-900-786-7200). There will be a charge of \$2.00/minute. Electronic support will continue through BIX.

SAS/C Compiler for AmigaDOS. Price: \$300.00. SAS Institute, Inc., SAS Campus Drive, Cary, NC 27513-2414, (919) 677-8000. Inquiry #225

HIGH DENSITY DRIVE

A relatively new entry in the Amiga market, **Applied Engineering** recently announced a new line of Amiga peripherals. Among them is the **AE High Density Drive**, the first and only high-density drive for the Amiga.

The AE HD (High Density) Drive supports both standard 880K and 1.52 meg 3.5" disks. Perks include a 2-way LED indicator that displays green for reading and red for writing, and smooth electronic disk ejection to replace that rather unrefined "punch" common to drive ejectors. And since it is tied to the write/enable function, the AE HD

Drive's electronic ejector waits until a drive completes a write to automatically eject disks, thereby preventing you from accidentally trashing disks.

The drive also features complete daisy-chain capacity (from AE to another drive, or vice versa), MS-DOS compatibility, pass-thru connectors, as well as a full one-year warranty.

AE High Density Drive. Price: \$239.00. Applied Engineering, P.O. Box 5100, Carrollton, TX 75011, (214) 241-6060. Inquiry #224



COOL.

The entire *Amazing Computing* library is now available at incredible savings of over **50%!**

AC Volume 1 is now available for just **\$19.95*!**

(A regular \$45.00 value, this first year of AC includes 9 info-packed issues.)

AC Volumes 2, 3, & 4 can be yours for just **\$29.95* each!**

(Volumes 2, 3, & 4 include 12 issues each and regularly sell for \$60.00 per volume set.)

VERY COOL.

And subscribers can now purchase freely redistributable disks** at distribution prices, so stock up!
This offer includes all Fred Fish, AMICUS, and AC disks. Pricing for subscribers is as follows:

1 to 9 disks: \$6.00 each
10 to 49 disks: \$5.00 each
50 to 99 disks: \$4.00 each
100 disks or more: \$3.00 each
(Disks are \$7.00 each for non-subscribers.)

To order volume sets, freely redistributable disks, as well as single issues, use your Visa or MasterCard and

call 1-800-345-3360.

Or just fill out the order form insert in this issue.

*Postage & handling for each volume is \$4.00 in the US, \$7.50 for surface in Canada and Mexico, and \$10.00 for all other foreign surface.

**AC warrants all disks for 90 days. No additional charge for postage and handling on disk orders. AC issues Mr. Fred Fish a royalty on all disk sales to encourage the leading Amiga program anthologist to continue his outstanding work.

MORE FOR YOUR MONEY

Commodore is now equipping the A3000, A2500/30, and A2630 card with some enhanced features.

Effective as of July 30, 1990, the A3000/16 and A3000/25 now come standard with a 50MB hard drive instead of the previously included 40MB hard drive. Commodore has also made the A2500/30 and A2630 card a tad more appealing by adding an additional 2MB of 32-bit RAM to both. With these changes, the A2500/30 now comes standard with a total of 5MB of RAM, and the A2630 card comes standard with 4MB of RAM.

Most important, these enhancements come at no additional cost to the consumer.

For more information, contact: Commodore Business Machines, Inc., 1200 Wilson Drive, West Chester, PA 19380, (215) 431-9100. Inquiry #222

ELECTRONIC CANVAS

Dakota Corporation, which distributes a wide range of graphics tablets for the IBM and Apple markets, is now exclusively distributing the **SketchMaster** graphics tablet for the Amiga. Designed for all graphics applications, the SketchMaster features a lightweight pen-stylus and does not require a power supply (it is powered directly by the RS232C serial port). Thus, a very natural, clutter-free working environment is provided for artists and designers.

SketchMaster works with all mouse-driven software and is fully compatible with IBM computers. The entire package includes a high-performance tablet, pen-stylus, 4-button cursor, interface cable, 9 to 25-pin adaptor, Dakota tablet driver, and manual. SketchMaster is available in two

sizes (11.7" x 11.7" and 12" x 8"), and is backed with a 5-year limited warranty.

SketchMaster. Prices: 11.7" x 11.7": \$449.00, 12" x 8": \$699.00. Dakota Corporation, 235 West Road, Portsmouth, NH 03801, (603) 427-0100. Inquiry #223

A LEARNING EXPERIENCE

Well, it seems McGee's insatiable curiosity has led the mischievous character to the Amiga. **McGee and Katie's Farm**, the first two installments in **Lawrence Productions'** "NO WORDS" software series, were recently made available for the Amiga. It is the aim of this particular series to allow children (mainly those in the 2 to 6-year-old range) to become acquainted with

computers through programs that, you guessed it, use no words.

In McGee, users can, through a simple point-and-click interface, travel with McGee through the house during the wee hours of the morning rousing the cat, Fido, or Mom. When McGee visits cousin Katie, in Katie's Farm, there's a whole new set of situations, including impromptu riding lessons and high-strung squirrels.

Both programs feature outstanding graphics and sound, and are designed to reinforce object/shape recognition, eye/hand coordination, cause and effect, and spatial relationships—all in a non-competitive, pressure-free manner.

McGee, Katie's Farm. Price: \$39.95 each. Lawrence Productions, Inc., 1800 South 35th, Galesburg, MI 49053, (616) 665-7075. Inquiry #226

•AC•

Explore Virtual Reality!



Step into the picture with Vista's camera. What you set is what you get.

VISTA



Crater Lake National Park

- Yosemite
- Crater Lake
- Mt. St. Helens
- Mons Olympus (Mars)
- 4 Billion Imaginary Fractal Landscapes
- Creates Frames for Animation
- Intuitive Interface
- Landscape Featuring Control

1 meg required • 2.0 compatible • List price \$99.95

Virtual Reality Laboratories, Inc.

2341 Ganador Court, San Luis Obispo, CA 93401 805/545-8515

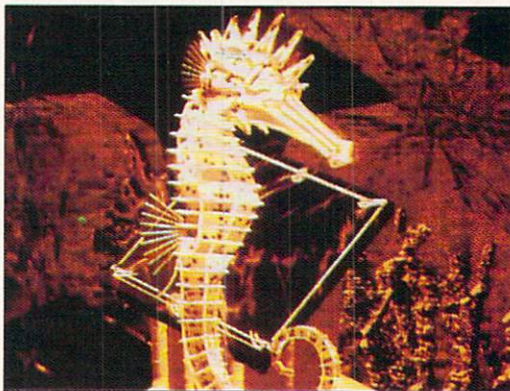
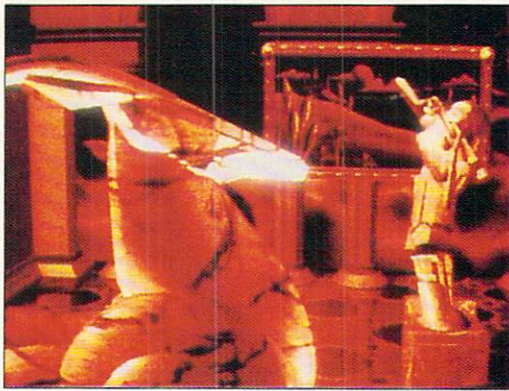
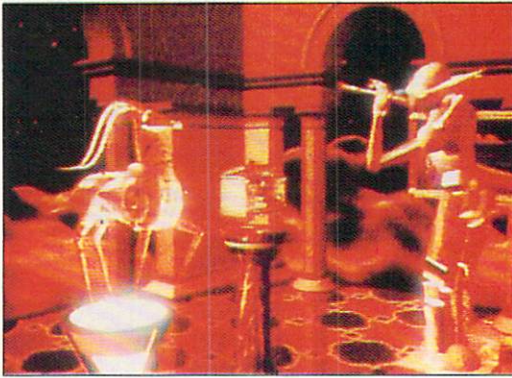
Reality prime will never look the same again!

Circle 131 on Reader Service card.

FrameGrabber 256

by Frank McMahon

Images grabbed off videodisc and digitized using FrameGrabber 256



*P*ROGRESSIVE PERIPHERALS & SOFTWARE has come out with a new version of their popular FrameGrabber digitizer. FrameGrabber 256 lets you digitize red, green, and blue video signals and features a new "MonoHAM" mode which allows any Amiga to display 256 shades of grey scales. If you are familiar with the original FrameGrabber, setting up the 256 will be a snap (the hookup is exactly the same).

The FrameGrabber 256 connects to any model Amiga through a standard Centronics printer cable. Again, that cable is *not* included with this package, a definite negative in my book. The fact is—if it's needed, it should be included. The package does come with everything else, in addition to the unit itself: external power supply, monitor cable, multi-purpose cable adaptor, software, RGB color filter wheel, and manual.

Connecting the digitizer is pretty easy. After hooking it up to your Amiga through the Centronics cable (which is actually a printer cable), disconnect your RGB monitor cable from your monitor and hook it into the FrameGrabber 256. Next, use the supplied cable and lead it from the FrameGrabber 256 to your monitor. An adapter comes with the package which allows you to hook the cable up to just about any monitor, including Commodore's 1080, 2002, 1084S, and 1084SD. One requirement that might be easily overlooked is an RGB monitor. You must have an RGB monitor for the setup to function properly.

The last hook-up would be a video source, inputted via the front panel. There are 4 RCA inputs

“... the FrameGrabber 256 software is miles ahead of most in ease of use, logic, and professional image enhancement.”

in the front of the unit. They can be used for standard composite video input from a VCR, camera, or videodisc. They can also be used to input separate red, green, blue, and sync signals. On the front panel next to the 4 RCA video inputs are two control knobs for adjusting the black level and intensity of the incoming video signal.

It's worth noting here that when the FrameGrabber 256 is hooked up, the unit must stay on as long as your computer remains on, even if you're not currently using the digitizer. This is because your RGB monitor signal is actually running through the digitizer itself. My original FrameGrabber has remained on for 1000's of hours in the last two years, so this should not be of too much concern.

However, the quality of your RGB monitor's signal is decreased going through all the extra hardware. This may not be noticeable under normal circumstances, but the slight blur does become noticeable when working with certain fonts in hi-res, for example. You may find it worthwhile to swap cables and bypass the FrameGrabber 256 when it is not being used for extended periods of time.

The manual includes no tutorials, but dives right into explaining what every menu option does. That's especially unfortunate because this program packs many more features than the original FrameGrabber. While the original allowed you to jump right in and start grabbing frames, this one really needs a good basic tutorial to describe the settings and different display modes. There are many excellent options from which to choose, and a brief walk-through would have been most helpful.

Previewing your incoming video signal is as easy as hitting the spacebar. You see the incoming video

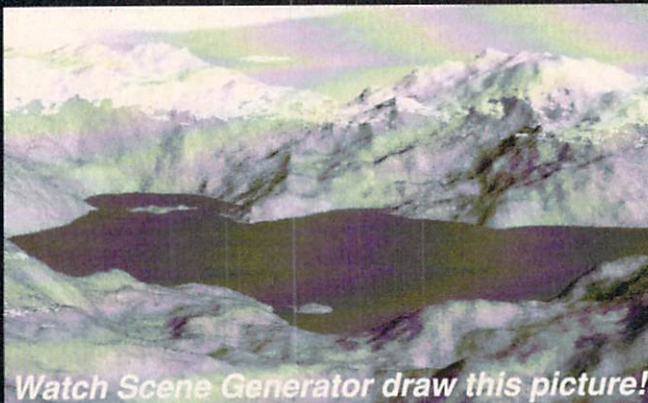
through your Amiga monitor in black-and-white. To grab an image simply hit the [C] key for "capture". Images are grabbed quickly, considering all the calculations the program performs to create quality output. While lo-res images can be grabbed in about 3 seconds, higher resolutions may take 2 to 3 times as long, which is still pretty fast. Images can be grabbed in several resolutions, including lo-res and interlace, in regular and overscan, using 2, 4, 8, 16, 32, 64, 128, or 256 colors. It is even possible to digitize in 24-bit mode for 16 million colors, although the Amiga cannot currently display 24-bit images without adding specific hardware (such as a framebuffer).

HAM mode is also available. Hit the [TAB] key to invoke the MonoHAM mode, which converts the entire screen into a black-and-white HAM mode with 256 shades of grey scales. This looks a lot smoother than the standard 16 shades of black and white we have come to expect with most Amiga digitizers.

The best way to view the different features of FrameGrabber 256 is through the various menus. One excellent feature is that nearly every menu command has a keyboard equivalent. This speeds up productivity quite a bit.

The Project Menu lets you load and save digitized images, plus pick up a part of your image and save it as a brush to use in a paint program.

Unless you have a parallel port-controlled printer, printing your finished product is quite easy. Since the digitizer uses the parallel port, it must be disconnected in order to plug your printer in. Of course, you can use parallel switch boxes, but don't feel bad if you don't have a box. Even though switching cables is laborious, it actually is the safest method to practice, as long as you make sure your computer is turned off before



Watch Scene Generator draw this picture!

Now you can create realistic, natural looking scenery on your Amiga with **Scene Generator**. The above picture is an example of one of the millions of scenes that may be created with this powerful new graphics tool. **Scene Generator** uses fractals to create natural scenery based on random numbers. You can change the steepness, snow and water levels, lighting angle and colors. Create everything from a desert to a snow covered mountain with lakes. The possibilities are nearly unlimited!

Available at your dealer. For credit card orders call (916) 624-1436 now. Or send \$49 to Natural Graphics, POB 1963, Rocklin, CA 95677. Free shipping USA.

Circle 128 on Reader Service card.

doing so. Parallel switch boxes can possibly damage a piece of hardware, as this manual rightly points out. It's best not to use any products that have not been specifically designed and/or recommended by a manufacturer.

One feature I like is the "Set Area" command in the print menu. It allows you to set an area with the mouse that you would like to print, so you do not have to print an entire graphic.

There is also a command in the project menu to turn on joystick control. Using an ordinary joystick enables you to grab frames and preview as well as grab red, green, and blue images using various joystick directions. The settings menu allows you to save all your variables like format, level settings, window positions, palette locks, etc. I was especially interested in Palette locks because I sometimes piece together presentations using AmigaVision or other slideshow programs. Locking the palette helps create smooth transitions during various wipes and dissolves because the presentation program doesn't have to keep switching color patterns. Being able to save my locked set of colors, I can go back anytime and digitize a piece of video and smoothly incorporate it into an existing presentation.

The Grab Menu allows you to grab red, green, blue and/or combined composite video (A color wheel is also provided with the package). This menu also

activates the MonoHAM mode as well as letting you choose what input you want to grab video from. Instead of using the four inputs as red, green, blue and sync you could have four different video inputs from four different sources continuously come in. Function keys F1-F4 allow you to instantly call up what video source you want to grab next! Automatic options such as AutoGrab, are also available. This allows the FrameGrabber 256 to constantly grab frames as rapidly as possible. AutoAnim also lets you continually grab frames as quickly as possible to create animations. We'll get into animating in a little bit but it is important to note that FrameGrabber 256 will not capture pieces of moving video in real time (such as the "LIVE" digitizer from A-Squared). It captures pretty quick, but because there is a certain amount of processing involved, the best you could do is capture images every 3 to 5 seconds or so. In fact, full timer options are included so you could set up your digitizer to capture plants growing or a sun setting and be able to play it back as a standard animation!

The Image menu allows you to show your current image and to set up the Format Control Window. This window is important because you will need to set your resolution and color amount ahead of time. In addition you can choose pseudo color, mono mode, or RGB mode. Dither is also fully adjustable with 3 presets (from none to full) as well as a requester that allows you to input a numerical dithering amount. You can choose from different sizes to digitize in (from full screen to postage stamp), as well as create a full screen pattern or "video wall paper" of several different images. These can also be animated to pop up one at a time. This creates an appearance that some one is constantly laying down strips of different video images in rows across the screen. The effect is quite impressive. The built in timer will capture images at certain intervals or constantly keep capturing until it reaches the amount of frames you have pre-specified. I created several of these animations and they loaded into Deluxe Paint III with no problem. They also can be played from within the program.

Other options in the Format Control Window are: border, mirror, timestamp (just like your camcorder!), oversampling, multiexposure, weighted, autozone, and cropdisp. These are basic enhancements that are implemented at the time of the digitizing. Delta is another option of the Image menu. It is a powerful processor that displays the "differences" between captured images. You may have heard of "delta-compression", which is a standard ANIM function which only plays the changes from frame to frame or image to image. This can save huge amounts of memory in animations, or for presentations where only a certain part of the screen needs to move. Delta lets you hunt down these changes to locate defects, detect motion, and eliminate unchanging background image clutter. Sequencer is used to automatically switch between the various video input on the front of the FrameGrabber

256. There are display options which allow you change your digitized image to any size or position on screen. You can also, from the Image menu, turn your pointer off if you are going to be videotaping your captured images. Jump Center is a sort of automatic centering that can also be controlled by the four cursor arrow keys for exact locations.

COMPLETE COLOR CONTROL IS AVAILABLE through the Color Menu.

The palette is always available and I'm glad to see options such as copy color, exchange color, and color spread. The Find command in the palette window allows you to choose a color and have the color flash in the digitized graphic. It provides an excellent way to find a certain shade if you are working with many similar hues. There is also a Color Bias window which enables you to change the brightness, contrast, and the saturation of colors. Playing with the palette and color bias window gives you the option to create hundreds of different color and intensity variations. Next in the Color menu is the Histogram window, one of my personal favorites. It features a graphic representation of the grey level intensities of the current image, as well as a host of powerful image processing options. There are commands such as Flat, which provides better contrast in the dark and light areas of the image, but loses some of the mid-range gray scales and Neg, which instantly converts the image to a photographic negative. Sharp is an edge image enhancement that does wonders for improving the sharpness of the final image. Average does the opposite, in that it blends the adjacent pixels together giving a smooth watercolor feel. Edge is an edge detection process which creates sort of a line drawing of the original image. Next to the Histogram is a set of Level controls for changing everything from the delta difference to the saturation.

I've worked with numerous graphic programs and can honestly say that the FrameGrabber 256 software is miles ahead of most in ease of use, logic, and professional image enhancement. The end results can be positively stunning.

The Anim menu we touched on and there are basically controls to set a timer to grab frames at certain intervals. You can also manually grab frames by hitting the capture key. All frames are automatically added to your animation and you can end the animation process at any time or set a frame amount before hand. A playback command is also provided to view your Anim. The Alarm mode gives off an audible tone before it digitizes a frame, which helps in stop motion animation.

CONCLUSIONS

The screen shots that accompany this article were some of the first things I digitized using FrameGrabber 256. They were grabbed off videodisc using the over-scan interlace setting. I first grabbed them in the MonoHAM mode and then decided to try to load them



Great Prices! Shipping based on weight and zone.
For Information & Catalog Call
Voice: 414-544-2066
Pursuitable BBS: 414-544-6567

At Safe Harbor, there are just a few things we say "No" to:

No club or membership fees!
No hidden charges!
No broken promises!
No hassles!
No nonsense!

You don't have to take our word for it. Just ask our customers! They'll tell you. At Safe Harbor, we're changing the way you think about mail order.

No foolin'!

Orders Only Please:
800-544-6599
Visa/MC/CODs

Circle 134 on Reader Service card.

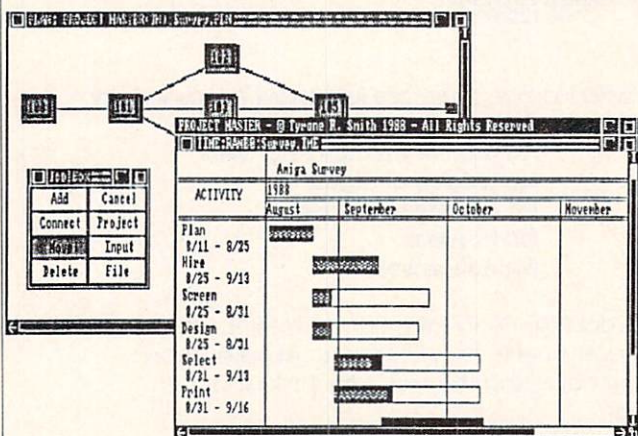
2414 Pendleton Place ■ Waukesha, WI 53188 ■ 9 AM to 5 PM M-F

into Photon Paint. To my surprise, they loaded with all weird colors and couldn't be remapped to look normal. Now I'm all for new formats like MonoHAM but I can't help but feel the same as I did when DigiView came out with hi-res HAM ... what do I do with it?

About all I can do is grab them and look at them. It's too bad they can't be used (yet) in other programs or multi-media presentations. After I disconnected the FrameGrabber 256 to hook my printer back up, I decided to load my digitized pics into the program so I could save them to another disk to send off to *Amazing Computing* for this article. Another surprise ... you need the FrameGrabber 256 connected and running to get into MonoHAM mode to display your 256 grey scale images. Now, we're talking "limiting". This excellent mode is limited to only those who own the digitizer. Eventually what I did was re-grab the frames, using only 16 colors of grey scale. To my surprise, with the advance dithering techniques and options such as sharpen, they looked almost as good as the MonoHAM images! After I grabbed them, I decided to change the normal black-and-white color spread to a dark brown-and-gold color spread, to give an added textured feel. There is currently no support for hi-res, Progressive states that it is unfeasible due to lack of adequate color bandwidth, increased memory demands, and economic considerations. Unfortu-

Project Master

Project Management for the Amiga



TRSL
P.O. Box 2746
New Orleans, LA, 70176
1-504-524-6404



Published by Brown-Wagh Publishing, Inc.

Circle 130 on Reader Service card.

nately, black and white digitizers really NEED the hi-res mode, and it was kind of a let down that hi-res wasn't available with this current software/hardware combination.

Aside from these problems, everything eventually comes down to quality, and that of the FrameGrabber 256 can't be beat. It allows you to capture real-time moving images with crisp quality, and also comes with an arsenal of high-powered image-processing tools; in short, Progressive's new digitizer is one that is going to be hard to top. Digi-View has always has the market cornered with the 3 primary color method (red, green, and blue), and now Progressive has implemented that same method into it's digitizer. Add to this capabilities for grabbing moving video from camera, VCR, videodisc, in up to 16 million colors—AND animating it! Progressive Peripherals & Software has come out with a product that, despite some shortcomings, will become their second digitizing hardware standard in the Amiga video world.

FrameGrabber 256
 Progressive Peripherals & Software
 464 Kalamath Street
 Denver, CO 80204
 303-825-4144
 Price: \$724.95
 Inquiry #216

•AC•

The "Videojitter"

Anyone who has digitized has seen it. It's out there, and it's bound to crop up once in awhile. No, it's not some wacky dance craze that your parents keep hoping comes back. It's a bigger problem.

You'll be okay if you stick to lo-res when grabbing frames with the FrameGrabber 256, the original FrameGrabber, or any digitizer that captures frames of moving video. But once you move into the interlaced resolutions, you will notice that if you hit capture, what you reel in may be a shaky, wobbly video image.

It's no fault of the FrameGrabber. It has to do with the way the video signal is constructed. Every video image is made up of frames. When you slow down a video tape you can see that, like film, it is made up of a series of frames. If you REALLY slow the video tape down (assuming you have a good four-head VCR), you will see what appear to be frames, but are actually fields. A frame is made up of two fields, which are interwoven together for higher apparent resolution. In fact, every other field will seem a little higher, because it is. It's only one scan line higher, though, so don't feel too bad if you don't notice. The two fields are quickly scanned to create a frame.

The problem arises from the fact that when a digitizer grabs a frame from video, it's actually grabbing two images ... two fields, which make up a frame. And if there is any motion going on in the frame, you'll get two fields which are not identical, causing a "wobble" or "jitter" of sorts.

The solution? Well, there isn't really one. A TV station—with thousands of dollars of equipment—grabs a freeze frame, and the same thing happens. They usually revert to grabbing a field which is about the equivalent of a lo-res screen.

So, if you are grabbing video with a lot of motion, stick to lo-res. Or if you have a VCR or a videodisc player that offers a solid freeze frame, it would be the best route to go. In fact, in grabbing pics for this article, I had to use the freeze frame on my videodisc ... too much "video jitter" otherwise. —Frank McMahon

Letter-perfect Graphic Accessories

KARA Fonts

by R. Shamms Mortier

KARA FONTS. ANYAMIGA ARTIST AND/OR VIDEO-OBSESSED INDIVIDUAL who is not familiar with these beauties is either new to the Amiga experience, or has had their eyes closed for the duration. The KARA Fonts disks have been around for a while, and can be seen in use in a multitude of formats, everything from desktop-published pieces to instructional and broadcast videos.

KARA Fonts fall into an Amiga category known as "ColorFonts", a trademark of the company that invented the Calligrapher font design program. The concept was (and still is) so original and useful to Amiga folks that Electronic Arts incorporated the ability to access

some time. All of the Amiga packages that I consider worth their purchase price can manipulate these font sets. They look great in black and white as desktop publishing output, and really shine for video applications.

All of the KARA Font sets are created in hi-res. There are ways, however, to use them in any Amiga resolution (including HAM). Just be aware of what different screen resolutions do to pixelated graphics. For instance, a video-res screen (320 x 400) is half the width of a hi-res screen (640 x 400). This means that images created in hi-res and imported into video-res will look twice as long horizontally as they would on their original hi-res screens. Adjustments must be made to the KARA characters if you want to see them in their pristine clarity. HAM screens come in two non-interlaced

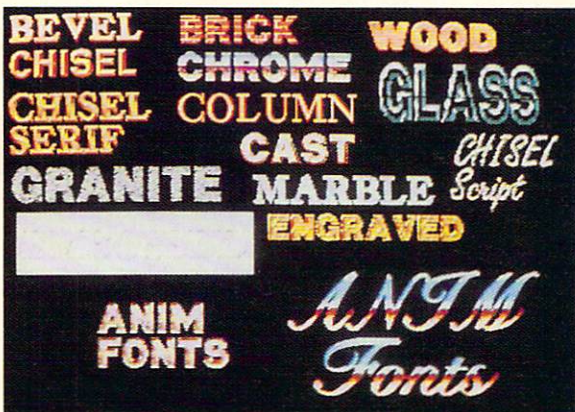
flavors: lo-res (320 x 200) and video-res (320 x 400). We've already suggested a way to treat the latter. lo-res is even simpler. KARA Fonts loaded onto a lo-res screen can be reduced by half to attain a resemblance of their original selves (though lo-res jaggies do not do the fonts

any favors). My favorite way to incorporate KARA Fonts into video-res HAM mode is to treat them first in a standard paint program like DPaintIII. First I compose my words, then grab them as a brush and reduce them by half. Then I take the half-sized brush and double its vertical dimensions. I import the saved text brushes into my HAM paint program, and voilà! They are in the exact proportions as the original KARA words.

The ability to outline a brush in DPaintIII does wonders with these fonts. One technique I use is to pick up my text after it is composed in a full brush chunk. Next, I use the outline function to place a thin border around each brush. This can be used as is to pop the brush out of a background color, or by choosing the "line" tool, to create great drop shadows. I also experiment with the colors of the font sets. It's not difficult to change one metallic look to another, as most are just color-based. Bizarre effects can be achieved by setting your palette to multi-colors and painting the characters down. There is also a list of alternate color combinations available as brushes on the KARA disks. Resizing the brushes to half size in hi-res doesn't hurt them a bit. They're as sharp as ever, giving you much more compositional space in which to shape the look of your message.

These fonts are a natural for video animation, especially reacting with grace and kindness to the manipulations one can direct in DPaintIII. Using the "MOVE Requester" in DPaintIII to put a KARA Font text brush through its paces never fails to give incredible results. I always turn on the maximum anti-aliasing

(continued on page 28)



A sample of what awaits the owner of the KARA Fonts multiple disks. In addition to several disks of extraordinary fonts, two ANIMfont flavors are available.

Colorfonts in their DeluxePaintIII program. It's so easy to speak of something being a "standard" in the world of micro-computing, but most of the time that term is just hype. Not so with KARA Fonts. They are a standard indeed, and will probably remain so for quite

Gradient Color Dithering on the Amiga Made Easy

by Francis M. Gardino

COLOR DITHERING IS THE SPATTERING OF 2 OR more colors so that the pixel colors optically mix to form additional colors. When an Amiga artist fills an area of a picture with a dithered pattern, we call that a dithered fill. When the colors of a dither pattern are in a gradient (i.e., progressing rather smoothly from one color to the next), we call that a dithered gradient fill. Within the five popular Amiga paint programs, there are various dithering features that give you a multitude of dithered gradient fill effects.

DeluxePaint III, the Amiga's flagship paint program, has a very simple, straightforward approach to gradient fill dithering. The dither control panel can be accessed by clicking on the fill tool with the Right Mouse Button or by the F keystroke. This fill requester, although it has its limitations, can provide you with the tools to create some interesting effects. The coarseness and smoothness of the dither can be controlled by the slider bar at the bottom of the panel. The direction in which the dither fills the area can be controlled by selecting one of the three arrow gadgets above the slider bar. The color range (for example, from red to purple) can be dithered from the top to bottom edge, or from the left to right edge, in the area to be filled. The fill can also be dithered from the left to right edge to follow contours, thereby achieving a

more rounded appearance in circles, ellipses and freehand curves. This contour fill will also follow the direction of diagonal straight line areas as well as boxes and polygons.

After creating and applying various dither fills to a picture, you can then alter them further by the use of the smearing, smoothing, blending, and shading effects in the pull down menu under Modes. By

Fig. 1: Some dither options available in DigiPaint 3.

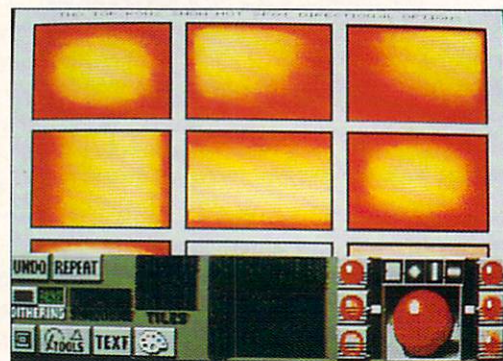
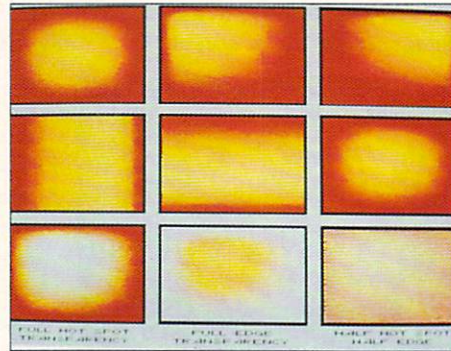


Fig. 2: Control requester shows Hot Spot and transparency controls on the right.

Dither options in DeluxePaint III and DigiPaint 3— and how PixMate can help to merge the effects of both

selecting the freehand tool and the largest brush in the tool box, you can smear and smooth until there isn't anything left to smooth.

In concert with these controls, DeluxePaint's color palette requester (keystroke p) allows you to control the range of colors used in the dithers, the number of colors, and the RGB value of any color within that range. The ranges and dithered gradient fills of DeluxePaint can have up to 64 colors, although you can only control 32 of them since the 32 Extra-HalfBrite colors are simply darker versions of the first 32 colors in your palette. Clicking on the arrow above the OK button reverses the direction of the colors in your gradient fill.

As you experiment with the various dither patterns and colors, you will notice that the colors used in gradient fills are as important to the smoothness or coarseness of your fill, as is the dither pattern selected in the Fill requester. Color ranges which have colors with 2 common RGB values throughout the range, only changing the remaining value by one increment, seem to have the smoothest results. For example, the top six gradients in Fig. 1 have the 16 RGB values R15,G0,B0, R15,G1,B0, R15,G2,B0....R15,G15,B0. At this point, DeluxePaint III is the program of choice for precise control of each RGB value in a range to achieve these results.

DigiPaint 3, a HAM paint program, does not give you precise control over every color in your dithered fill, but it does, however, give you far greater control of the various dithered effects and directions for use in your fills. The many complex dithering features of DigiPaint provide numerous options from which to

(continued on page 29)

Fig. 3: DigiPaint 3 background exported via PixMate to DeluxePaint, where the spheres were added.

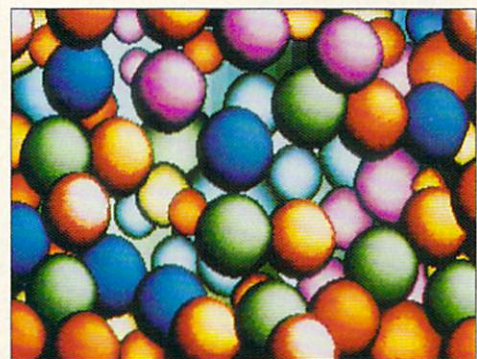
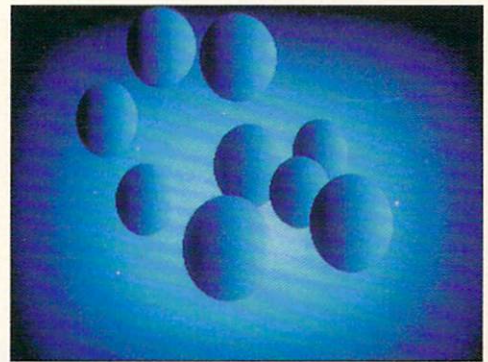


Fig. 4: A hybrid image created using DigiPaint's gradient fill pattern and DeluxePaint's RGB color controls.

Sculpt Script Tutorial

by Christian Aubert

SCULPT-ANIMATE 4D IS A PRETTY powerful three-dimensional modeler and renderer. The graphic interface is easy to use but, at the same time, somewhat limiting. Powerful scripting commands built into this package, on the other hand, give you total control over the finished product.

A steep learning curve for these commands results in a situation similar to that of Workbench vs. CLI: one is easy to use but has its limitations, the other is powerful but more difficult to implement and master.

Nevertheless, I have started using Sculpt's (we'll call it this way from now on) scripting commands. The fact that they are only briefly noted at the very end of the manual makes them seem like something usable only by the very seasoned Sculpt user.

If I haven't scared you off already, then you are in for a lot of fun. Let's start off by designing a familiar object that is quite simple to render—

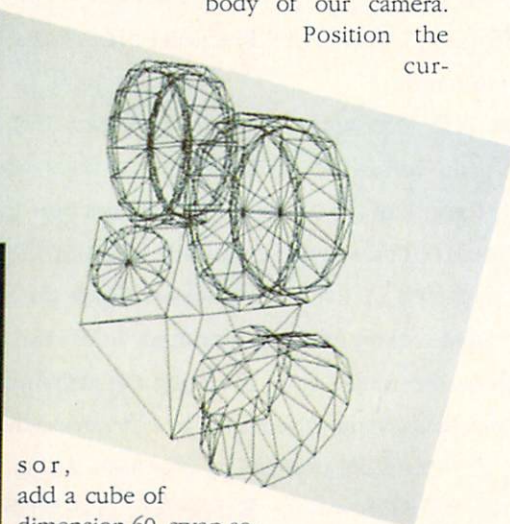
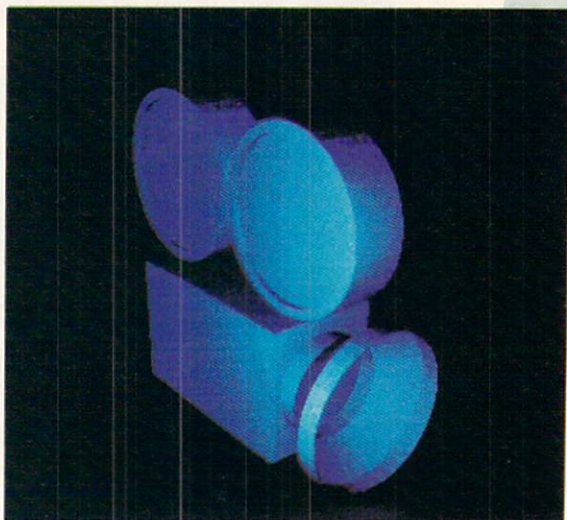
an old-style movie camera with the twin film reels on top.

Line 1 makes the West window the active one; we need to make sure of the plane we are working in for the DISK, CIRCLE, and SPIN commands. Try designating the NORTH or the UP window as the active one, and you will see the difference. Line 2 discards the sky. Now we are ready to start our design.

Lines 3 through 19 make up the lens. Position the cursor, add a circle of radius 40 with 16 subdivisions, select the circle, reposition the cursor, extrude to the next position, and ungrab the circle. Repeat the extrusion again, and contract the circle in the Up-Down/North-South directions. We then repeat the extrusion, select everything and name it "LENS".

Lines 21 through 26 make up the body of our camera.

Position the cur-



sor,
add a cube of dimension 60, swap selected vertices so we end up with the cube selected, expand it in the Up-Down and East-West directions and name it "BODY".

Lines 28 through 39 make up the battery cell at the back of the camera. Position

"Powerful scripting commands built into this package...give you total control over the finished product."

the cursor, add a disk of radius 25 with 16 subdivisions and make sure that only the disk is selected. Extrude it to its new position and name it "BATTERY". Ungrab the object, deselect all and erase the center vertex of the battery, which is hidden by the body of the camera and only adds more faces and complexity to the object (read: longer rendering times and more memory usage).

Lines 41 through 48 make up the first reel. Make a T shape, select it, position the cursor at the feet of the T and then spin it 360 degrees in 16 subdivisions. Select it and name it "REEL1".

Lines 50 through 54 make up the second reel. We select only "REEL1", make a duplicate of it, and move the duplicate to a new position. Ungrab it and name it "REEL2".

Lines 56 through 60 set the properties of color and texture for the whole scene.

Finally, lines 62 through 72 set the resolution, viewing mode, and size of the image; determine position of the observer and target; add a lamp, and start the rendering process.

The result is a camera model that can be used in other creations. If you want a more detailed model for up-close viewing, change the number of subdivisions in the CIRCLE, DISK, and SPIN commands to higher values. Do the opposite if you want a small object to appear in the background, or if memory considerations oblige you to show less vertices and faces.

With a little planning and imagination, you can get some pretty impressive results, and we have only scratched the surface of this language. I leave it up to you to discover more!

```
WINDOW WEST
SKY NONE
(0,0,40)
ADD CIRCLE 40 16
SELECT ALL
(0,0,40)
EXTRUDE
(10,0,40)
GRABBER OFF
EXTRUDE
(30,0,40)
GRABBER OFF
EXPAND 0.65 UD
EXPAND 0.65 NS
EXTRUDE
(40,0,40)
GRABBER OFF
SELECT ALL
NAME SELECTED VERTICES "LENS"

(95,0,40)
ADD CUBE 60
SELECT SWAP
EXPAND 1.16 UD
EXPAND 1.83 EW
NAME SELECTED VERTICES "BODY"

(158,0,40)
ADD DISK 25 16
SELECT ALL
DESELECT NAMED VERTICES "LENS"
DESELECT NAMED VERTICES "BODY"
EXTRUDE
(150,0,40)
NAME SELECTED VERTICES "BATTERY"
GRABBER OFF
DESELECT ALL
(150,0,40)
```

```
ERASE SELECTED VERTICES

WINDOW WEST
(52,11,120)-(52,11,160)-(52,14,160)-(52,14,165)-
(52,-14,165)
(52,-14,160)-(52,-11,160)-(52,-11,120)
ELECT CONNECTED
52,0,120)
PIN 16 360
ELECT CONNECTED
NAME SELECTED VERTICES "REEL1"

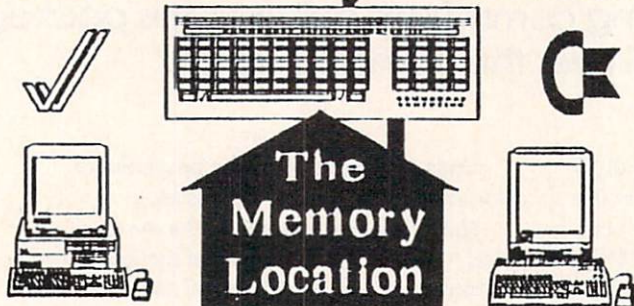
ADD DUPLICATE
GRABBER ON
(142,0,120)
GRABBER OFF
NAME SELECTED VERTICES "REEL2"

SELECT ALL
TEXTURE SHINY
SET TEXTURE
FACE COLOR 0.4 0.45 1.0
SET FACE COLOR

HIRES
INTERLACE
WIREFRAME
IMAGESIZE FULL
(-210,230,240)
LOCATION
(-150,230,140)
ADD LAMP
(83,0,75)
TARGET
START
```

•AC•

Come see whats Hot for the **AMIGA** at The Memory Location



396 Washington Street
Wellesley, MA 02181
(617) 237-6846

**AMIGA Experts! Nothing but the best.
Satisfaction guaranteed.**

**Come in and try-out all the latest software,
hardware, and accessories.**

Your full service *AMIGA* dealer.

**Store hours: Mon.-Thur. 10-6 Friday 10-8 Saturday 9-5.
Full Commodore authorized repair service onsite.
Low flat rate plus parts.**

Circle 107 on Reader Service card.

(Kara Fonts, continued from page 23)

function to smooth out the edges. A fast-moving animated brush, however, will hide its jagged diagonals, especially when dumped to NTSC for broadcast. (I never let a client view Amiga graphics on an RGB monitor, but choose a composite or NTSC monitor instead. That way, they get the real idea of the broadcast results.) Another trick I've used in the past is to take my composed text and paint it down on several animation frames. I then use the "Smear" tool in DPaintIII to mess the edges on several frames in succession. When played back, you can achieve a look that comes off as explosive edges being generated. You could use a similar technique to make KARA Fonts appear to be condensing out of a chaotic environment, or vice versa. Another thing to remember about Amiga animation techniques is to use color cycling with KARA Font compositions. The sets are so

smoothly colored, and the color ranges are so evenly distributed, that color-cycled KARA Fonts take on a special glow and sparkle. I have even used sections of a KARA Font letter as a fill for other shapes, because the fills are so well executed.

AND NOW, PRESENTING...

Kara Computer Graphics is not a company that rests on its laurels. In addition to their list of fine video-font libraries, they've decided to produce sets of ANIMfonts. These come in two varieties, and the incorporated fonts can be used in standard non-animated ways as well. The first set writes itself on the screen like a signature. Since this is a script-style font, it is well-suited to the animation procedure used. The second animated font set has a square, metallic look. It can be made to appear on the screen by going through a vertical turn

(on the Y axis). Each would be useful for major video-titling projects. The directions for getting the fonts to jump through their hoops are clearly notated in the accompanying document sheets. Experienced Amiga users will probably already be able to guess how the process works. It's simple enough. The separate letters and symbols are saved as ANIMbrushes. One thing the docs don't mention is that ANIMbrushes can be flipped and manipulated in other ways, so all of the possible options here are not indicated. Experimentation is heartily suggested. Although it is suggested that these fonts can be used with a long list of step-by-step referenced programs, I would strongly urge purchasers of the KARA libraries to become familiar with DPaintIII. This seems to me to be the most useful Amiga animation program to utilize these fonts. The ANIM fonts require at least a meg of RAM, and that means CHIP RAM.

LOWER-CASE LONGING

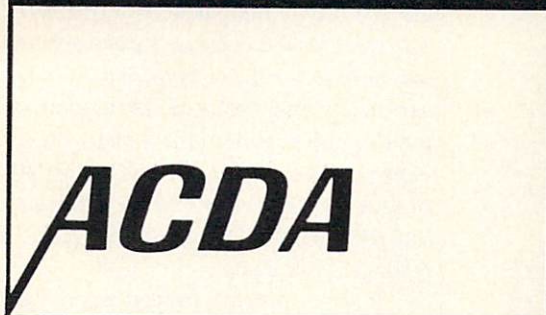
I have only one concern about the KARA libraries, which has been with me for as long as I've used these fonts: Where are the lower-case letters? Yes, the script fonts have lower-case equivalents, but not the other sets. It is a fact of graphic design that it is more difficult to read words made up of all caps than it is to discern upper-case/lower-case combinations. This becomes even more true when a message is longer than two words. I am hoping that Kara Computer Graphics is working to bring lower-case libraries to market. When they do, I'll be the first order on their list.

•AC•

Headlines (3-disk set): \$79.95
HeadLines 2 (2-disk set): \$69.95
SubHeads (2-disk set): \$69.95
AnimFonts 1: \$49.95
AnimFonts 2: \$49.95
Kara Computer Graphics
2554 Lincoln Blvd., Suite 1010
Marina Del Rey, CA 90291
(213) 578-9177
Inquiry #214

Simply put, ACDA provides the only complete line of Scientific and Engineering enhancements for your Amiga computer.

1. High performance data-acquisition(A/D) and process-control(D/A, DIO). The DataStation™ products.
2. GPIB IEEE-488.
3. DigiScope and other signal analysis software tools.
4. Laserdisk Control Software.
5. Hardware and Software Consultation.
6. Unsurpassed quality and support.



ACDA Corporation

220 BELLE MEADE AVE, SETAUKET NY 11733
Tel. 516/689-7722
FAX 516/689-5211

*Amiga is a trademark of CBM, Inc.

Circle 104 on Reader Service card.

(Color Dithering, continued from page 25)

choose. One such option is the ability to dither-gradient-fill a line.

The directions of the dithers in the control requester are similar to DeluxePaint, although not entirely the same, with some very unique and interesting modes and effects available (see Fig. 1). There is the standard left-to-right-edge and top-to-bottom-edge gradient fill, but there are also what they call Hot Spot controls. These allow you to specify the location of the highlight of your gradient fill. By sliding the Vertical or the Horizontal Hot Spot control bar, you can create a gradient that goes from red to purple and then back to red again, with the highlight defining the turning point. This may seem like a small feature, but it is not so easy to replicate in some of the non-HAM paint programs.

The circle Hot Spot control allows you to specify the location of the round highlight as it is applied to all shapes, not merely circles or ellipses (see Fig. 2). The transparency slider bar to the left of the

above mentioned Hot Spot control, determines the transparency of the Hot Spot, that is, the amount of the Hot Spot color that the gradient fill will contain in its highlight.

The slider bar to the right of the Hot Spot controls determines the dither's degree of edge transparency to the color below. When the highest position is selected, the outer edge colors of the fill, as well as the inside, are opaque. Conversely, when at the lowest position, the edge colors are at the most transparent setting. HAM paint programs do this by combining the applied colors with the existing picture colors to form new colors without any regard for exceeding the 16-, 32- or 64-color barrier of the non-HAM paint programs. These calculations have a palette of 4096 colors from which to choose and display. To access these range color effects, the Range item must be selected under the menu item Modes.

The selection of colors to be used in a color range are accessed in the color palette requester (click on palette tool). Only the first and last colors of the range can be changed by selecting the boxes at either end of the range, then using the Pick color gadget or the RGB sliders in the center of the requester. The intermediate colors of the range are computed by DigiPaint. Most of the time, these intermediate colors transitions are smooth and pleasing to the eye, but sometimes they are not. Occasionally the blends computed from one color to another, like pink to red, go by way of an unexpected color, like green. This lack of control of the intermediate colors can be overcome by importing brushes and pictures from another paint program like DeluxePaint, where one can control all of the colors used in the desired range.

The dither patterns and smoothness controls have two gadgets each, located in

Attention

Readers!



AC WANTS YOU!

Since we at Amazing Computing™ cannot determine the dependability of our advertisers from their ads alone, we want your feedback. If you have had a problem with an advertiser in AC™, let us know! Send a complete description of your exchange(s) with the advertiser—along with the names of the individuals involved—and we'll do our best to get to the bottom of things.

Ad Complaints
PiM Publications, Inc.
Amazing Computing
P.O. Box 869
Fall River, MA 02722

the same DigiPaint control panel over the labels "Smoothing" and "Dithering". Herein lies another one of the limitations of DigiPaint 3. There are no slider bars to control the dither pattern or smoothness. The four gadgets control it all. Despite this lack of control, the blend possibilities remain numerous and varied.

As in DeluxePaint, DigiPaint allows you to further alter the appearance of these dithers by the use of the features included in the menu items under Modes. There lies Darken, Lighten, Colorize, Blur, RubThru, TxMap, And, Or, Xor and the many possibilities of using these effects with varying degrees of Edge and Hot Spot Transparencies. Unusual and sometimes unpredictable effects are possible when exploring this area of DigiPaint. These effects can be applied to the image after certain dithers are created, since to create a Range dithered fill, one must have the Range mode selected, under the same menu item Modes.

Through the magic of programs like PixMate and the IFF standard, you can export some DigiPaint effects to DeluxePaint, with predictable results. Incidentally, the new 24-bit IFF standard allows you to import a 32-color brush or picture into the 24-bit image, only taking up 32 of the colors available in your new palette. So, these images should still be of use to the 24-bit artist.

Generally, if you try to simply load a HAM-produced picture into a non-HAM paint program, you will either get a message saying that it can't be done or some other unpredictable result, even if you have only a few colors in your HAM picture. With PixMate, however, you can convert a 16-color DigiPaint image (such as the one in Fig. 3) to a picture usable in DeluxePaint, by following this procedure:

First, paint a filled shape in DigiPaint using the various dither, Hot Spot, etc. controls, and specify a 16-color range from R15,G15,B0 to R15,G0,B0. Even though

these may not be the colors you want to use in your final image, it's important to use them now for the conversion process. You can go into DeluxePaint later to get the actual colors you want. Save your HAM image, load it into PixMate and pull down the HAM-to-32-color conversion command. When you load this 32-color image into DeluxePaint, the first 16 colors will appear as black, and your 16-color palette will fill up the last two rows of the palette.

Returning to PixMate, you may want to make one small change after converting and before resaving your image. Go into the PixMate palette and change the second color to white (remember, all the others in the first half of the palette will be black). This controls the color of the gadgets and menus in DeluxePaint; if you left this color as black, you would not be able to see these tools on the screen. At this point, you can either make color adjustments in PixMate, or save the 32-color IFF image and load it into DeluxePaint to adjust the gradient colors to your liking.

It also is possible to load a non-HAM picture file into DigiPaint and the other HAM paint programs without any noticeable side effects, other than some color fringing. Once saved in the HAM paint program, however, the file becomes a HAM file which must then go through the PixMate procedure mentioned above to return to a non-HAM format.

As you can see, by importing, exporting, converting formats and making small adjustments to your color palettes, you can enhance the quality and variety of your images (see Fig. 4). Experimentation with the controls in these programs can provide the user with hours of inspiration and joy, as well as visually pleasing results.

•AC•

ABOUT THE AUTHOR:

Fran Gardino has a Bachelor of Fine Arts from Mass. College of Art and is the Art Director for the Boston Computer Society's Amiga Culture newsletter. He is also the developer of Ham It Up!, a color utility for the Amiga.

Acting on Impulse

by John Steiner

IT WAS A SULTRY JULY AFTERNOON WHEN I PULLED INTO THE PARKING LOT OF building 6870 on Shingle Creek Parkway in a suburban industrial complex a few miles north of Minneapolis. I had heard that Impulse was preparing to release a 24-bit graphics display board for the Amiga, and I wanted to check it out in person. This was the place to do it.

Professional graphic artists have frequently complained about the Amiga's limited array of only 4096 colors. The Firecracker 24 board breaks the 4096-color barrier, as it is capable of delivering up to 16.8 million colors to either Amiga RGB- or NTSC-compatible monitors.

I had telephoned Impulse and spoken with company President Mike Halvorson, and he invited me to stop by and take a look. Once the car was parked, I grabbed my 35mm camera and entered the lobby. The office directory told me where to find Impulse; a few turns down the hallway, and I was there. After introducing myself, I was invited into a workshop that sported several Amiga systems, most of which were in some state of disassembly. I could see that these units were left open to permit easy removal and reinsertion of cards as they are being tested.

A tall, bearded young man was introduced to me as Mr. Halvorson, and he welcomed me by asking me to have a seat in front of a caseless Amiga 2500. The 1084 monitor which the Amiga was connected to showed a Workbench screen that appeared to be genlocked over a high-resolution video still image, the source of which I could not readily determine.

Halvorson told me that I was looking at an image that was being displayed via the Firecracker 24. He explained how the

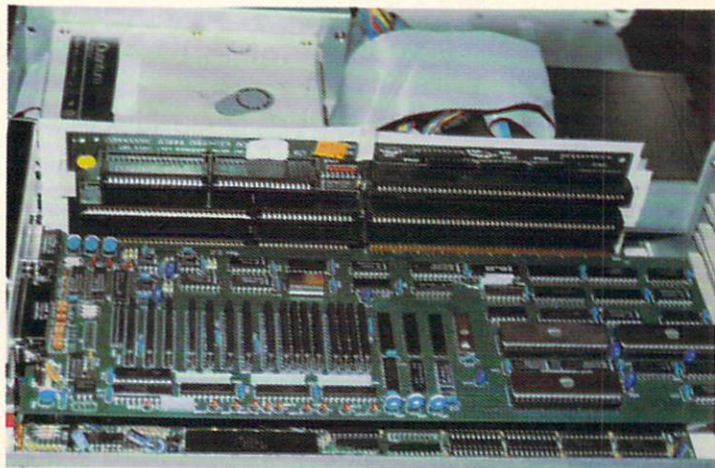
video port signal from the Amiga 2500 was first passed through the Firecracker board, then out a standard Amiga video connector to the 1084.

He explained that this internal genlocking capability serves two purposes. First, the end user is able to view the 24-bit images and Amiga Workbench without requiring an extra video display monitor. Secondly, software developers are able to use the standard operating system calls when writing software for the Firecracker board. Clarified, this means that the interface makes it easy for a developer to support the Firecracker board in a specific program. Because of the unique design of the Firecracker 24, programmers won't have to rewrite major sections of program code so that their programs can display images using this new board.

Mike went on to say that the Firecracker board supports the recently released

24-bit IFF image standard recommended by Commodore. Software that can already be used with the Firecracker 24 includes Turbo Silver, Imagine and The Art Department, among others. He also commented that the board, which is currently being evaluated for FCC approval, will ship with a 24-bit paint program called "Light". The board will be released for sale upon completion of the certification process. It may even be shipping by the time you read this.

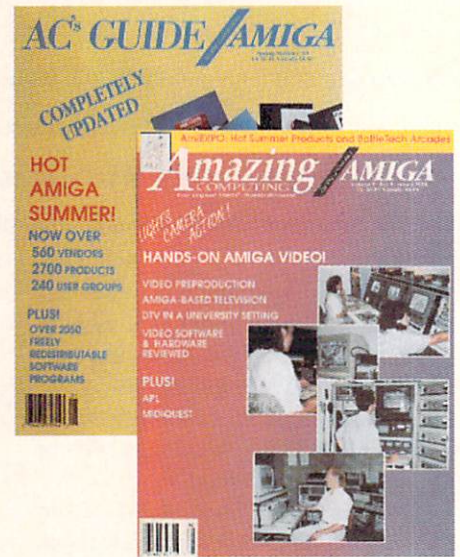
The 24-bit images I saw were displayed with an image viewer program developed in-house by Impulse. The pictures were nothing short of stunning. Mike handed me a specification sheet which provided some more information about the board. According to the sheet, the standard resolution of the Firecracker 24 with 1 megabyte of RAM on board is 512 x 482 pixels. The 2-megabyte board can deliver up to 1024 x 482 pixels.



Pick a Number!

Save over

49%



1-Year Savings

- 12 issues of *AC* for just \$24.00*
Save over \$23.00 off the newsstand price!
- *SuperSub* \$36.00*
12 issues of *AC*,
Plus 3 issues (1 year) of *AC's Guide*!
Save over \$32.00!

Even with these great numbers, there is still one number that is most important to us. **Our readers are always Number 1.** Every issue of *AC* is designed to provide a maximum amount of

Any Number!

Save over

59%



2-Year Savings

- 2 years of AC for just \$38.00*
Save over \$56.00 off the newsstand price!
- 2-year SuperSub \$59.00*
24 issues of AC,
Plus 6 great issues of *AC's Guide*!
Save over \$77.00!

Amiga information. That's why AC has always been Amiga users' **Number 1** magazine.

Here's the last number you'll need:

1-800-345-3360

*foreign prices differ

Amiga video system users should note that the Firecracker 24 is genlockable to the Amiga as well as to other video sources through the use of a standard Amiga



Viewing 24-bit images and the Amiga Workbench with no extra video monitor needed.

genlock. The Firecracker board takes up one of the normal Amiga slots in either a 2000 series or 3000 computer, and does not occupy the video slot. Halvorson mentioned that there may be a conflict with certain types of genlocks that use the video slot, which could cause problems. He went on to point out that they have tested the board with several different genlocks. Their engineers can provide further information on possible conflict avoidance through their technical support lines, for those who are considering the purchase of a Firecracker 24.

Once he whet my appetite for spectacular color graphics on the Firecracker,

Halvorson went on to demonstrate Impulse's new real time 3-D image editor, Imagine. Impulse is probably best known for their premier 3-D software package,

Turbo Silver, one of the best 3-D modeling packages for any computer platform. Imagine makes 3-D object creation much simpler than any other 3-D software package I have ever seen. I don't often admit to my own deficiencies, but I have never been able to comprehend the three-view interface found in most 3-D packages, and my feeble attempts at 3-D object creation have always ended in failure.

When I first saw the standard three-view screen, I assumed that the Imagine interface would be as difficult and incomprehensible as the rest of the 3-D modeling programs I've worked with. I couldn't have been more wrong. Mike started the demonstration by using a pull-down menu choice to create a sphere, which appeared in all three views—top, side, and front. A perspective view was also visible on the screen.

Mike quickly zeroed in on the top view and started moving control points about. He then moved to the front view and moved some more points. To my amazement, the perspective view was taking on

the appearance of an alien robotic head. In moments he had taken a primitive graphic and created something that looked as if it might be useful in my own video work.

He commented that Imagine is not a rework of Turbo Silver, but that it is new from the ground up. Some of the better features of Turbo Silver have been included, but both the editor and animation system are completely new.

There are two object editors, a three-view point wire frame editor that Mike had been using, and a detail editor that can not only sweep, spin and extrude objects, but can also drill, gouge, join, extract and slice them. There is even a magnetism feature that can "pull" points toward the magnetic pointer.

The animation cycle editor built into Imagine does not involve any complex script language. Mike created a "stick man" from previously made cylinders. Once the stick man was created, Mike positioned the stick man in a walking pose, and identified that pose as key frame 1. He then repositioned the stick man, and identified the new pose as frame 2. The animation generation process defines all the positions in between the two key frames, and automatically creates the animated image. From this stick figure, you can create true key cell operations called cycles. Examples are walk cycles, run cycles, dance cycles or any other repetitive movement.

Within a few moments, the stick man was walking. He mentioned that Imagine can "morph", or change one form into another. You can also animate object attributes, thus changing the color or texture of an object in an animated sequence.

Mike concluded the demonstration by creating a simple black-and-white logo in a 2-D painting program. The logo was saved in IFF format as a single bitplane (two color) image. He brought the 2-D image into Imagine's 3-D editor where it could be extruded into a 3-D image, then colored, textured and animated.

Mike commented that Imagine would be shipping by the end of August and that registered Turbo Silver owners would be able to buy the program for \$150, for 60 days from the release date or date of purchase of Turbo Silver, which is a \$200 discount off the regular retail price. Regis-

AMIGA • AMIGA CHIPS, PARTS & UPGRADES • AMIGA			
•Fatter (Super) AGNUS 8372 - \$99.50 with simple 10 minute step by step instructions, FREE Chip/Extractor valued at \$9.95 at no charge.			
• 8520A CIA	\$17.95	• 8373 Super Denise	CALL
• 256 x 4/100 ns	\$9.30	• 2.0 Kickstart ROM	CALL
• 1 meg x 1/100 ns	\$9.40	• A501 512K Upgrade	\$74.00
• A500 Keyboard	\$109.50	• Keyboard for A1000	\$129.95
• A500 Service Manual	\$36.00	- AND MANY OTHERS -	
• A1000 Service Manual	\$29.95	• 8362 Denise 1/2 bright	\$39.95
		• 1 x 4 meg/80 Zip (A3000)	\$59.95
		• A2000 Keyboard	\$114.95
		• A500 H/D Pwr. Supply	\$69.95
		• A2000 H/D Pwr. Supply	\$147.00
		• A2000 Service Manual	\$39.00
• AMIGA 1000 REJUVENATOR PACKAGE •			
New Product - The Amiga 1000 Expansion Board is now available with the following features: Utilizes the Fatter Agnus Chip, 1.3/1.4 Kickstart ROM and New Denise • One Meg of Chip RAM • Clock-Battery Backup • Simple Solderless Installation • 100% Compatibility with all Products/Software • Various Packages Available • \$479.00 complete.			
		The Grapevine Group, Inc. 35 Charlotte Drive Wesley Hills, NY 10977 1-800-292-7445	
Prices subject to change		• (914) 354-4448 FAX (914) 354-6696 Write for complete catalog.	
Circle 147 on Reader Service card.			

MOVING SPECIAL!

To serve our growing customer base, we are moving into larger quarters!

MEMORY

8-UP Memorycard-2MB	259.95
8-UP Memorycard-4MB	379.95
8-UP Memorycard-6MB	499.95
8-UP Memorycard-8MB	549.95
Starboard II for A1000 with 512K	199.95
with 1MB	249.95
M501 512K upgrade for A500 with clock	69.95



System Packages

We customize AMIGA 2000 and AMIGA 3000 desktop video systems to meet your individual needs. Call and talk to one of our system specialists to get the best price for your requirements.

GVP 68030 - 28 / 68882 / 4MB
1899.95

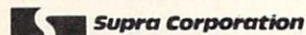
GVP 68030 - 33 / 68882 / 4MB
2549.95

GVP 68030 - 50 / 68882 / 4MB
3699.95

GVP

DRIVES

FData-10 Single drive with on/off	99.95
FData-20 Double drive with on/off & power	239.95
AIR drive external	109.95
Master 3A	109.95
Internal 3.5 drive for A2000	89.95
Cutting Edge MAC drive for use with AMAX	179.95
Quantum Hardcard 40MB	499.95
Quantum Hardcard 80MB	799.95
HardFrame SCSI Controller	199.95
GVP SCSI Controller with 8MB space	199.95



For the AMIGA 500:	
Supra 40MB Quantum	599.95
Supra 80MB Quantum	849.95

For the AMIGA 1000:	
Supra 40MB Quantum	649.95
Supra 80MB Quantum	899.95

Supra Wordsync card	149.95
---------------------	--------

SOFTWARE

3D Professional	299.95
Turbo Silver	89.95
Disney's Animation Studio	119.95
Saxon Publisher	249.95
Pagestream 2.0	179.95
The Art Department	59.95
Imagine	219.95
Scene Generator	29.95
Digi Paint	64.95
Pro Write 3.0	109.95
Pro Video Post	219.95
Mega Paint	179.95
Deluxe Paint III	89.95
Homebuilder's CAD	119.95
Lattice 5.0	189.95
Disk Manager MAC	79.95
Soundscape Pro MIDI	109.95

ICD ADRAM 540

with 4MB & 501 trade-in	349.95
without 501 trade-in	399.95

VIDEO

Digi View 4.0	129.95
Digi Works 3D	79.95
Color Splitter	119.95
Flicker Fixer	469.95
Framebuffer with 1MB	749.95
Magni 4004S w/ remote	1629.95
MiniGen	179.95
Neriki Imagemaster II	1699.95
Panasonic 1410 w/ lens	199.95
Panasonic 1500 w/ lens	299.95
Panasonic Vari-Lens	39.95
Polaroid Freezeframe	1599.95
ProGen	349.95
Sharp JX-100 Scanner	749.95
SuperGen	669.95
SuperGen 2000	1549.95
Vid Tech Scanlock	899.95
VIP Video Interface	99.95

*Government and School
Purchase Orders Accepted.*



InterComputing, Inc.

2112 Sandy Lane, Dallas, TX 75220 • Customer Service: 214-556-9666 • FAX: 214-556-2336

NEW ORDER LINE!

1-800-800-9177

InterComputing Deutschland Inc.

Schönebecker Str. 55-57 Telefon: 0202/89155
5600 Wuppertal-2 Telefon: 0202/89304

InterComputing France

34, Avenue des Champs Elysees
75008 Paris

Phone: (1) 42821603
FAX: (1) 42806649

As always we have the most 'customer friendly' terms: S/H \$4.95 in cont. USA; \$30.00 min. order; MASTERCARD & VISA with NO credit card fee; in Texas add 8.25% Sales Tax. \$12.00 shipping to APO/FPO addresses. RMA# required on all returns. All prices subject to change.





So once again, you probably spent too much money on your summer vacation this year.

Right?

Well, there is something you can do to replenish your bank account in time for that next vacation—write for *Amazing Computing!*

Do it! Share your hot new ideas with thousands of other serious Amiga users just like yourself in the pages of AC, and earn some cold, hard cash in the process!

Just call us toll free at 1-800-345-3360 (1-508-678-4200 in MA) to discuss your ideas with one of our editors.

We'll steer you in the right direction, then mail you a packet of information to help you get started as an author.

And once you get published—we'll kick off your 1991 vacation fund!

tered users will have been notified of the release of the program, so it's a good idea for you to register your Turbo Silver software if you haven't already done so.

I took a few minutes to ask him about his involvement in the Amiga Developers Association. Then, Mike had another appointment to keep, so he introduced me to Michael Dammar, the Director of Research at Impulse, and the designer of the Firecracker 24. Mr. Dammar answered a few more of my questions regarding the board's self-genlocking capabilities, and helped me set up my 35mm camera to take a few pictures. We finished that, and it was time for me to head back to Fargo.

The infectious enthusiasm of Mr. Halvorson and Mr. Dammar rubbed off on me, and I left with the feeling that the Amiga will survive as long as dedicated companies and individuals such as these continue to make high-quality, professional products for Amiga owners.

•AC•

Firecracker 24

Impulse, Inc.

6870 Shingle Creek Parkway #112

Minneapolis, MN 55430

(612) 566-0221

price undetermined

Inquiry#200

The Amiga Developers Association

Mike Halvorson has been involved with the Amiga development community for some time. Over the years, he has seen a need for a much more organized approach toward working with Commodore, and hastened development of Amiga hardware and software support. Mike decided to bring his ideas into a tangible form; thus, the concept of an Amiga Developers Association was born. The ADA was created with two major goals in mind: to facilitate communications between the Amiga software and hardware developer community and Commodore, and to strive for improvements in the way the Amiga is marketed and promoted. Mike felt that the developers needed an organization that could maintain complete independence from Commodore, and he enlisted the help of Doug Barney of AmigaWorld magazine to help put the organization together.

The Amiga Developers Association became a reality, and Mr. Halvorson was elected President of the ADA at its first meeting. With the new management at Commodore fully supporting the new organization, the ADA has found that many of its initial goals have already been achieved. The group is now changing its focus to enhance acceptance of the Amiga in specialized business applications.

Halvorson's already exuberant personality began to show even more enthusiasm as he talked about how the Amiga is really a great communications tool. He personally likes to refer to the Amiga not as a computer, but rather, as a "digital graphic imagination machine"! Communicating with imagery bypasses any language barriers, and Halvorson believes that developers and dealers should work to push the Amiga into new, previously untapped market areas and business applications.

He went on to say that everyone from Commodore itself to the smallest dealers are still marketing the computer to the wrong people. Dealers need to market to those who are probably already using PCs to fulfill many of their business needs, but who *also* need the Amiga for what it does best—graphics communication.

For example, the color graphics and sound generated by the Amiga are a lot more attractive than the LED "crawling" signs found in many retail stores, and believe it or not, an Amiga 500 color graphics display system can be much less expensive.

Halvorson feels that everyone needs to take advantage of the inherent creativity and marketing potential built into the Amiga.—J.S.

AC Disks

Source code and executable programs included
for all articles printed in *Amazing Computing*.

#1 AC V3.8 and V3.9

Gels in MultiForth Parts I & II: Learn how to use Gels in MultiForth. Author: John Bushakra

FFP & IEEE: An example of using FFP & IEEE math routines in Modula-2. Author: Steve Fawiszewski

CAI: A complete Computer Aided Instruction program with editor written in AmigaBASIC. Author: Paul Castonguay

Tumbler Tots: A complete game written in Assembly language. Save the falling babies in this game. Author: David Ashley

VGad: A gadget editor that allows you to easily create gadgets. The program then generates C code that you can use in your own programs. Author: Stephen Vermeulen

MenuEd: A menu editor that allows you to easily create menus. The program then generates C code that you can use in your own programs. Author: David Pehrson

Bspread: A powerful spread sheet program written in AmigaBASIC. Author: Bryan Catley

#2 AC V4.3 and V4.4

Fractals Part I: An introduction to the basics of fractals with examples in AmigaBASIC, True BASIC, and C. Author: Paul Castonguay

Shared Libraries: C source and executable code that shows the use of shared libraries. Author: John Baez

MultiSort: Sorting and intertask communication in Modula-2. Author: Steve Fawiszewski

Double Playfield: Shows how to use dual playfields in AmigaBASIC. Author: Robert D'Asto

'881 Math Part I: Programming the 68881 math coprocessor chip in C. Author: Read Predmore

Args: Passing arguments to an AmigaBASIC program from the CLI. Author: Brian Zupke

#3 AC V4.5 and V4.6

Digitized Sound: Using the Audio device to play digitized sounds in Modula-2. Author: Len A. White

'881 Math Part II: Part II of programming the 68881 math coprocessor chip using a fractal sample. Author: Read Predmore

At Your Request: Using the system-supplied requestors from AmigaBASIC. Author: John F. Wiederhorn

Insta Sound: Tapping the Amiga's sound from AmigaBASIC using the Wave command. Author: Greg Stringfellow

MIDI Out: A MIDI program that you can expand upon. Written in C. Author: Br. Seraphim Winslow

Diskless Compiler: Setting up a compiler environment that doesn't need floppies. Author: Chuck Raudonis

#4 AC V4.7 and V4.8

Fractals Part II: Part II on fractals and graphics on the Amiga in AmigaBASIC and True BASIC. Author: Paul Castonguay

Analog Joysticks: The code for using analog joysticks on the Amiga. Written in C. Author: David Kinzer

C Notes: A small program to search a file for a specific string in C. Author: Stephen Kemp

Better String Gadgets: How to tap the power of string gadgets in C. Author: John Bushakra

On Your Alert: Using the system's alerts from AmigaBASIC. Author: John F. Wiederhorn

Batch Files: Executing batch files from AmigaBASIC. Author: Mark Aydelotte

C Notes: The beginning of a utility program in C. Author: Stephen Kemp

#5 AC V4.9

Memory Squares: Test your memory with this AmigaBASIC game. Author: Mike Morrison

High Octane Colors: Use dithering in AmigaBASIC to get the appearance of many more colors. Author: Robert D'Asto

Cell Animation: Using cell animation in Modula-2. Author: Nicholas Cirasella

Improving Graphics: Improve the way your program looks no matter what screen it opens on. In C. Author: Richard Martin

Gels in MultiForth-Part 3: The third and final part on using Gels in Forth. Author: John Bushakra

C Notes V4.9: Look at a simple utility program in C. Author: Stephen Kemp

1D Cells: A program that simulates a one-dimensional cellular automata. Author: Russell Wallace

Colourscope: A shareware program that shows different graphic designs. Author: Russell Wallace

ShowLBM: A program that displays lo-res, hi-res, interlace and HAM IFF pictures. Author: Russell Wallace

Labyrinth II: Roll playing text adventure game. Author: Russell Wallace

Most: Text file reader that will display one or more files. The program will automatically format the text for you. Author: Russell Wallace

Terminator: A virus protection program. Author: Russell Wallace

#6 AC V4.10 and V4.11

Typing Tutor: A program written in AmigaBASIC that will help you improve your typing. Author: Mike Morrison

Glatt's Gadgets: Using gadgets in Assembly language. Author: Jeff Glatt

Function Evaluator: A program that accepts mathematical functions and evaluates them. Written in C. Author: Randy Finch

Fractals: Part III: AmigaBASIC code that shows you how to save/load pictures to disk. Author: Paul Castonguay

More Requestors: Using system calls in AmigaBASIC to build requestors. Author: John Wiederhorn

MultiForth: Implementing the ARP library from Forth. Author: Lonnie A. Watson

Search Utility: A file search utility written in C. Author: Stephen Kemp

Fast Pics: Re-writing the pixel drawing routine in Assembly language for speed. Author: Scott Steinman

64 Colors: Using extra-half-brite mode in AmigaBASIC. Author: Bryan Catley

Fast Fractals: A fast fractal program written in C with Assembly language subroutines. Author: Hugo M. H. Lypkens

Multitasking in Fortran: All the hard work is done here so you can multitask in Fortran. Author: Jim Locker

#7 AC V4.12 and V5.1

Arexx Part II: Information on how to set up your own Arexx programs with examples. Author: Steve Gilmore

Leggo My LOGO: A Logo program that generates a Christmas tree with decorations. Author: Mike Morrison

Trees and Recursion: An introduction to binary trees and how to use recursion. Written in C. Author: Forest Arnold

C Notes: A look at two data compressing techniques in C. Author: Stephen Kemp

Animation? BASICally: Using cell animation with AmigaBASIC. Author: Mike Morrison

Menu Builder: A utility to help build menus in your own programs. Written in C. Author: Tony Preston

Dual Demo: How to use dual playfields to make your own arcade games. Written in C. Author: Thomas Eshelman

Scanning the Screen: Part four in the fractals series. This article covers drawing to the screen. In AmigaBASIC and TrueBASIC. Author: Paul Castonguay

C Notes: Recursive functions in C. Author: Stephen Kemp

#8 AC V5.2 and V5.3

Dynamic Memory! Flexible string gadget requester using dynamic memory allocation. Author: Randy Finch

Call Assembly language from BASIC: Add speed to your programs with Assembly. Author: Martin F. Combs

Conundrum: An AmigaBASIC program that is a puzzle-like game, similar to the game Simon. Author: Dave Senger

Music Titrer: Generates a titler display to accompany the audio on a VCR recording. Author: Brian Zupke

C Notes From the C Group: Writing functions that accept a variable number of arguments. Author: Stephen Kemp

Screen Saver: A quick remedy to prolong the life of your monitor. Author: Bryan Catley

#9 AC V5.4 and V5.5

Bridging The 3.5" Chasm: Making Amiga 3.5" drives compatible with IBM 3.5" drives. Author: Karl D. Belsom

Ham Bone: A neat program that illustrates programming in HAM mode. Author: Robert D'Asto

Handling Gadget and Mouse IntuiEvents: More gadgets in Assembly language. Author: Jeff Glatt

Super Bitmaps in BASIC: Holding a graphics display larger than the monitor screen. Author: Jason Cahill

Rounding Off Your Numbers: Programming routines to make rounding your numbers a little easier. Author: Sedgwick Simons

Mouse Gadgets: Faster BASIC mouse input. Author: Michael Fahrion

Print Utility: A homemade print utility, with some extra added features. Author: Brian Zupke

Bio-feedback/Lie detector Device: Build your own lie detector device. Author: John Iovine

Do It By Remote: Build an Amiga-operated remote controller for your home. Author: Andre Theberge

#10 AC V5.6 and V5.7

Convergence: Part five of the Fractal series. Author: Paul Castonguay

Amiga Turtle Graphics: Computer graphics and programming with a LOGO-like graphics system. Author: Dylan McNamee

C Notes: Doing linked list and doubly linked lists in C. Author: Stephen Kemp

Tree Traversal & Tree Search: Two common methods for traversing trees. Author: Forest W. Arnold

Exceptional Conduct: A quick response to user requests, achieved through efficient program logic. Author: Mark Cashman

Getting to the Point: Custom Intuition pointers in AmigaBASIC. Author: Robert D'Asto

Crunchy Frog II: Adding windows and other odds and ends. Author: Jim Fiore

Synchronicity: Right and left brain lateralization. Author: John Iovine

C Notes From the C Group: Doubly linked lists revisited. Author: Stephen Kemp

Poor Man's Spreadsheet: A simple spreadsheet program that demonstrates manipulating arrays. Author: Gerry L. Penrose

#11 AC V5.8 and V5.9

Fully Utilizing the 68881 Math Coprocessor Part III: Timings and Turbo_Pixel Function. Author: Read Predmore, Ph.D.

C Notes From the C Group: Functions supporting doubly linked lists. Author: Stephen Kemp

APL and the Amiga: Programming APL on the Amiga. Author: Henry T. Lippert, Ed.D.

Time Out! Accessing the Amiga's system timer device via Modula-2. Author: Mark Cashman

Stock-Portfolio: A program to organize and track investments, music libraries, mailing lists, etc. in AmigaBASIC. Author: G. L. Penrose

CygCC: An Arexx programming tutorial. Author: Duncan Thomson

Programming in C on a Floppy System: Begin to develop programs in C with just one megabyte of RAM. Author: Paul Miller

To be
continued ...

The Art Department 1.01

by R. Shamms Mortier

ONCE IN A WHILE, AMID THE FLOOD OF NEW AMIGA SOFTWARE, THERE APPEARS a "must have" package. ASDG's "The Art Department" (TAD) falls into that category, especially for those users involved in Amiga DeskTop Publishing and/or Video. With the pending release of Workbench 2.0 and the extensive chip upgrades, the Amiga may yet fulfill its promise as a leader in DeskTop Video (DTV), and as a competitive contender for DeskTop Publishing (DTP) honors. Utilities that help it address the technical and professional DTP and DTV goals are still needed in this venture, and ASDG hopes to be able to provide many of the solutions.

ASDG has a long and valued history as a primary supporter and developer of Amiga products. As such, they are in a good position to oversee and address the Amiga's future DTV and DTP needs. TAD (The Art Department) is the first of many anticipated new releases toward this end, and it is a downright superlative piece of software.

TAD OF MANY FORMATS

TAD is a graphic utility that translates and manipulates a host of diverse Amiga formats into standard IFF and multiple-bitplane files. It can import formats ranging from all standard IFF graphics in any resolution including HAM and overscan (with its "Super-IFF" loader; SHAM and A-HAM as well), DV21 files (older Digi-View), Digi-View 3.0 and Digi-View Gold, Rendition (used for Caligari Professional's 32-bit, 16-million color output "Broadcast Renderer"), Impulse (RGB8 files written by Turbo-Silver with 16 million colors in 24 bitplanes), Sculpt (the Mimetics 24-bit format); DeluxePaint II for MS/DOS (an IBM format with 256 colors), GIF (a CompuServe format that addresses IBMs and Apples with 1 to 8 bitplanes of data).

The Super-IFF, Digi-View, Sculpt 4D, and Turbo-Silver loaders come bundled with the software, plus additional modules can be purchased separately. ASDG claims it will support other loaders as needed in

the future, so the TAD software should be continuously upwardly modular.

TAD GOES TO WORK

TAD has an INSTALL option that installs it on either your hard drive or a bootable floppy. I installed it on a floppy, then also installed the Turbo Silver and Rendition loaders on the same boot disk. The interface is user friendly, elegantly designed, and the commands are logical and easy to comprehend. A one-time cursory reading of the manual is all that should be required for experienced Amiga users. Several sample pictures are stored on the original TAD disk for following the tutorials, however, installing the program on a bootable floppy or hard drive deletes them. They are, however, still importable into TAD from the original source.

At the top of the interface screen is a "LOAD FORMAT" selector which steps you through all of the loaders you have installed beforehand. You have to choose the one that coincides with the picture's format you are importing or it will not load. Next are the Color Controls for Balancing, Dithering, and calling up the Palette requester.

Balancing: Clicking on this gadget brings up a requester with many useful options. The manual gives a complete and detailed picture of exact functions of each of the balance sliders and suggests nominal

default settings. Here you can increment/decrement the Red, Green, Blue, Brightness, and Contrast of a loaded picture. There is also a slider named "Gamma" (whose default is set at 0, and which can be increased to 100). Gamma allows you to increase relative brightness/contrast of a picture without some of the anomalies associated with the brightness/contrast control alone. After setting any or all of these sliders, you hit the "Accept" button on the bottom of the requester. Finally, the "Execute" gadget on the main screen applies the transitions to the picture that is loaded. In a short time you are presented with the result. The nice thing about the "Balance" controls is that you can recover your original image without loading the picture again.

THE DITHER SETTINGS

There is a total of six dithering types in TAD, and those familiar with the Workbench 2.0 dither settings will no doubt recognize them: Floyd-Steinberg, Burkes, Sierra, Jarvis, Stucki, and Random. You can also select "None". Most of the dithering functions apply to DTP use, rather than to video applications, although some interesting effects can be created for video by experimenting with the various dithers. With no dither at all a HAM image suffers a bit in its apparent smoothness. Other resolutions, however, may actually be improved with this setting.

Palette: This brings up a standard Amiga palette requester with all of the controls familiar to experienced Amiga users. You can change individual colors, as well as setting smooth ranges of tones. There is one setting that I really appreciate: Sorting. When pictures are digitized, their palettes often have no smooth transition from one color to the next, making it almost impossible to change ranges of color. TAD, with the SORT option, sets up the palette colors from dark to light or vice versa. This requester also allows you to set colors aside

... ASDG is well known for their designed speed increases in hardware and software, and TAD seems to be the lucky recipient of all of their research and experience.

for genlock purposes, and to incorporate Workbench colors (for eventual display of the picture on the Workbench screen). Again, ACCEPT/EXECUTE puts the changes into action.

THE IMAGE CONTROLS

Your loaded graphic may be translated to "Line Art" (non-HAM images only), flipped vertically and/or horizontally, and converted to true grayscale (for non-IFF data). "RIP" ("Remove Isolated Pixels"), is the next gadget. It combs through a loaded image and selects those one-of-a-kind stray pixels that seem to infiltrate digitized visuals, and gets rid of them. Image Controls has an excellent "Scale" feature, which would do well to be incorporated into all Amiga graphics software. Width and Height parameters can be adjusted either numerically or by percentage sliders, to reduce or enlarge the loaded image. This attribute is worth the price of the software by itself. Upon activating the "Execute" command on the main screen, the graphic takes on the new dimensions.

SEPARATIONS

Besides the Load/Save/Display options, TAD has a gadget called "Separate" that should be invaluable to Amiga DTP users, especially those who wish to use HAM images in an application that demands 16 million color (or as 256 gray level) output. Colors may be written and separated according to any of three variations: RGB (the Red, Green, and Blue components familiar to microcomputer artists), CMYK (the Cyan, Magenta, Yellow, and Black separations common to printing professionals, also known as "four color process"), and CMY (a no-Black separation option, also known as "three color process"). Each can be saved to a depth of four or eight BitPlanes per color. Four BitPlanes would be used for Grayscale data (256 shades), and eight for full color (24

BitPlanes or 16 million registered colors). There are also sliders that allow both UCR ("Under Color Removal", indicating the amount of color to be removed with the addition of the "K" or Black Plane), and GCR ("Gray Color Replacement", indicating what percentage of the color removed will be added back as Black). These settings both have suggested default positions for the user not familiar with the technical details. Two more defaulted gadgets that allow for Magenta/Yellow Ink Compensation in the separations.

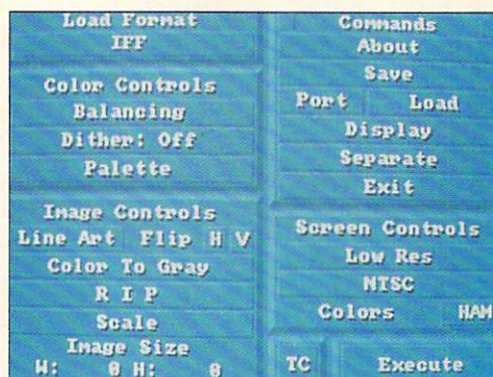
VIDEO USERS

So as not to leave the Amiga DTV user behind, TAD also addresses the MicroIllusion's Transport Controller software, allowing full 16 million color images to be transferred directly to video. Of course to use this, you have to be running MicroIllusion's Transport Controller software, and also be hooked up to a single frame controller and a high grade VCR. The manual offers many suggestions for using TAD's options in Amiga animation and graphic design applications.

Rounding out the controls are a series of toggles that allow the screen to be configured to accept Low-Res/Hi-Res, overscans, NTSC/PAL dimensions, and various color planes (from 2 to HAM, and including settings for EHB and A-HAM variations). A speed/time dial screen pops up to the front while your decision is being processed. Some processes are instantaneous, while others take a few seconds more. ASDG is well known for their designed speed increases in hardware and software, and TAD seems to be the lucky recipient of all of their research and experience.

TAD THE MIGHTY

An ASDG product called "ReSep",



This is the main interface screen of TAD.

which allows images translated by TAD to be printed out with ProPage as full 16 million color graphics, will be welcomed by Amiga DTP users of Gold Disk's Professional Page. 256 Graylevel images can also be output to a compatible printer.

TAD 2.0 will be released in the Autumn of 1990. In addition to loading all of the modular formats, it will also be able to save in the optional formats. This will increase its value and use geometrically. Many new modules are in development. This is qualitative professional software that needs to find its way into the hands of all Amiga DTP and DTV users (legally). The price of the software is far below its value for Amiga artists, designers, and animators. It has already become a standard tool in my studio, and I'm sure that it will serve you just as well.

•AC•

The Art Department: \$89.95

ReSep: \$59.95

Optional Loaders:

GIF: \$34.95

DPaint II Enhanced: \$19.95

Turbo Silver: \$24.95; Sculpt 4D: \$24.95

Targa 1,2,9,10: \$39.95

Rendition: \$49.95

TIFF: \$39.95; PCX: \$49.95

Inquiry #227

ASDG Incorporated

925 Stewart Street

Madison, WI 53713

(608) 273-6585

Scene Generator 2.10

by R. Shamms Mortier

MANY MONTHS AGO, I REVIEWED BRETT CASEBOLT'S SCENE Generator 1.0 on these very pages. Since that time, and due to the success of revising and marketing what was once a public domain program, Brett has written himself into the Amiga history books as an accomplished developer of an obsessive product. Brett's story might seem like another mystical Horatio Alger piece of Americana at first glance (you know, "small town boy makes good!"), but that's hardly the case. His story is really about dedicated hard work and the experience of disciplined creativity, as well as being at the right place at the right time. His company, "Natural Graphics", has used Scene Generator to develop other packages that Amiga obsessives will soon be enjoying as much as they do this package. But enough laudatory forecasting, let's get down to digital tacks.

A FRACTAL ENGINE

The idea behind Scene Generator owes its inception to none other than the renowned Benoit Mandelbrot and other fractal dimensional explorers. Fractal image generation takes on

a look and feel of the natural world because it gives a certain perceived order to chaos. Also, we seem to accept the self-similar reality we observe in fractal graphics as "real". Therefore, software that uses fractal techniques to generate imagery then produces images that are seemingly a cross between photographs and our day-to-day observations of reality. Many traditional artists see these techniques as non-art and even as threatening, in that they allow the machine itself to make certain decisions that have thus far been the prerogative and responsibility of the human artisan alone. Scene Generator may in fact interfere with the computer artist's longing to be recognized as a "legitimate" artist. After all, if the only intervention necessary in the creation of a work of art is that of plugging a "seed" number into an already resident algorithm and sitting back to watch the results, how can we call the outcome "art"? If you find yourself irate and insulted concerning artistic automata, then don't purchase Scene Generator. The results you perceive may throw you into an aesthetic quandary! In other words, it is definitely addictive.

This program produces some of the most inspiring "natural" graphic digital background paintings that the graphic designer and/or electronic painter will ever witness. The interface is so simple a two year old (or even a non-Amiga user) could interact successfully with it. You have a limited amount of light source and element choices, and an infinite amount of numerical "seeds". Taken together, your Amiga screen can display geography that resembles arid wastes, lagooned islands, stretches of seacoast, snow capped wilderness, and even planets on the outskirts of unknown galaxies. You can either stop after the scenes are generated and hang print-outs on the refrigerator, or you can export them to Amiga paint programs as luscious IFF and HAM backdrops. They also work very well as genlocked backgrounds for video applications.

The dithering routines in this software are so fine that the Lo-Res visuals look like HAM Video-Res paintings. Real Interlace can also be toggled, resulting in about twice as much time for generation, as well as more detailed images. Both the Lo-Res overscan and the Interlace overscanned images port to HAM painting software (MicroIllusion's



Scene generated in extra-high detail in about two minutes with Scene Generator.

. . . The results you perceive may throw you into an aesthetic quandary! In other words, it is definitely addictive.

Photon Paint or NewTek's DigiPaint). Overscan, by the way, can be set to normal or "severe", so the output can address professional video. Nine optional light source directions are possible. Each can radically alter a painting, moving it from a feeling of pleasantness and splendor to stark awesomeness. There is a selector for setting "detail" to Medium, Hi, or Extra-Hi. The latter is always my choice, for even though it increases the generation time, it offers the most qualitative results. This program is written entirely in Assembly language (which is Brett's area of expertise), so even the most detailed settings and resolutions render to the screen in a few minutes.

Water can be added to the scenes, and its perceived level set. Texture can be added to the water as well, and there is even a setting that allows "beaches" to be rendered automatically. The altitude can be input as data, giving you alternatives from a flat MidWest plain to sharp Himalayan peaks. In Vermont, our idea of scenery wouldn't be complete without an option for adding snow. Scene generator allows this to come to pass. Where the light hits the snow, it is a brilliant white, fading off into bluish shadow in areas that are opposite from the light source. The sky can also be set to receive wispy clouds, though I prefer to add my own skies later in a HAM paint program. Brett Casebolt has an addiction to mountain climbing and environmental issues, and no doubt all of this contributed heavily to his observations. These astute and appreciative observations are what fuels the internal beauty of Scene Generator.

Rendered scenes can be saved to disk in any of several formats: Settings alone; Settings plus non-overscanned picture; Settings plus overscanned picture; Settings plus severe overscanned picture (necessary for professional video work). "Settings" allows you to import a chosen scene's parameters back into the program. Brett is working on an animation utility as well, tentatively called "Scene ReAnimator". It will allow you to move around in the Scene Generator realm, and to save the moves as an ANIM file. Owners of Scene Generator will be able to upgrade to it at a comfortable price (release date and price not yet determined). Planned future versions will support accelerator cards with a separate program. An idea also in the realm of



Another Scene Generator background, customized with NewTek's DigiPaint 2.0.

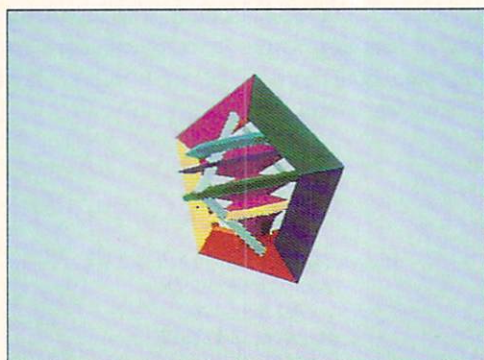
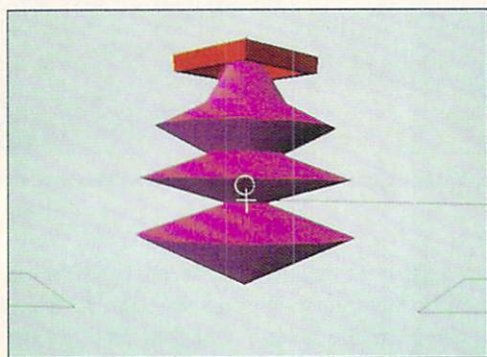
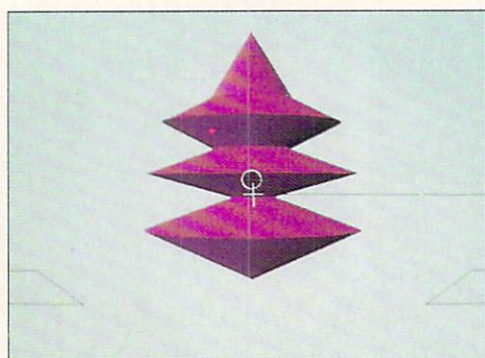
possibility is saving a scene as an object file, so that 3D rendering software (like Turbo Silver from Impulse or Sculpt 4D from Byte by Byte) can ray trace the terrain! I highly recommend this software. It is useful for Amiga artists and animators, and it is addictive to the max. Oh. One last thing. You'll probably want to save so many of your new Scene Generator creations that you will have to upgrade your hard drive, or purchase hundreds of floppies. Enjoy. See you in ROMulan space!

•AC•

Scene Generator (version 2.10)
Natural Graphics
PO Box 1963
Rocklin, CA 95677
(916) 624-1436
Suggested Retail Price: \$49.95
Inquiry #220

Breaking the Color Limit with PageRender3D

by R. Shamms Mortier



Figures 1 (top), 2 (middle), and 3 (bottom):
This series of pictures shows the development of the
PR3D multicolored hi-res object. Notice that more
colors increase the graininess of the dithering.

WHAT HAPPENED TO THE GOOD OLD DAYS (TWO years ago) when there wasn't so much great software to choose from? PageRender3D, one of an integrated series of packages from Mindware International, has a long list of exciting and useful features not offered in any other Amiga animation/3-D rendering package. Over two years in development, the attributes of PageRender 3D, and the results you can obtain with it, combine for an overall approach and impact unmatched by any other rendering/animation package that I have had the opportunity to work with.

The PR3D manual is just as complete as those of other Mindware products, and just as clearly written. PR3D has a lot of parameters, though, so be prepared to spend some time studying the techniques it uses. Mindware does not copy-protect its wares. The manual also has a complete index of terms and processes, and is spiral bound to fold out flat on your workspace. It has four main sections: an overview of program concepts, tutorials and reference material (the largest section), sample scripts, and the appendices. By the way, no other Amiga software developer that I know starts out their manual with a quote from the third Zen Patriarch, perhaps a promise of astounding transformations to come.

TOOLS AND ICONS: A DIFFERENT APPROACH

There are really two separate PR3D programs included on the two disks. One is the standard version, while the other, which runs significantly faster, is especially made for users with a 68020 chip. As far as being visually recognizable as compared to other Amiga software of the same or similar purpose, PR3D's tools will require you to memorize a whole new list of options. I hope to zero in on many of them in future Amazing articles. If you are familiar with the language and concept of PR3D's sister program, PageFlipper+FX, this should decrease the introductory time spent learning this program. Even the dragbar takes on a new shape and placement, allowing you to multitask with other software. There is the Gadget Window, the PR3D CLI scripting window (all commands evoked by the mouse can also be input in scripted form, and mouse commands also write the command out simultaneously in scripted notation in an INFO window, forcing you to learn the syntax), and a larger CLI type "console" window. Above all of this is the display

"... you can create some truly stunning multicolored graphics and animations."

screen and the drop-down menus, so you can readily see why learning to operate comfortably in PR3D takes a bit of study.

THE GADGET WINDOW

Here is a listing of some of the syntax equivalents:

GO – moves the position of the observer, moving the observer point and the view point simultaneously.

GORD – (short for "Go Around") moves the observer around the GORD origin, and keeping the view centered on that origin.

LOOK – Changes direction observer is looking.

MOVE OBJECT – Moves the work object.

MOVE CTR – Moves the center of the work object.

MOVE LIGHT – Moves the light source (there are six icons on the screen that tell you at a glance what kind of a center is involved in a move: A circle for the object's center, a double-cross for the view point, a diamond for a light source, open crosshairs for the GORD origin, and crosshairs for the coordinate system origin, and the observation point).

ROTATE – rotates objects around their rotation axis. A short pause and explanation is due here. Not satisfied with an XYZ Cartesian coordinate system alone, PR3D has other possible coordinate systems that can be employed in the animation process. Spherical, tetrahedral, and cylindrical systems are possible, and objects react quite differently to rotational and other commands, depending upon the coordinate method selected.

STRETCH – stretches the object by an indicated factor along any or all of the axis chosen.

CYCLIC SYMMETRY – here's where it really gets interesting. This command clones duplicates of an object around its X, Y, or Z axis.

MIRROR SYMMETRY – clones an object by interposing a mirror plane perpendicular to the selected axis.

REPRODUCE – clones the work object.

PAINT – brings up a nicely designed requester that allows user to select and alter colors of the palette (PR3D works in all Amiga resolutions, including HAM and overscan, so the number of colors are resolution dependent).

ERASE – removes selected object (this interfaces with several possible commands: object, pick object, pick facet, UNDO, and EXIT the erase mode).

DRAW – creates a picture using the setup and facet files in memory. It's best to turn it off when performing large numbers of alterations, then toggle it on again when done. Does not produce a ray-traced drawing.

BRACES – used for scripting in looped directions.

There are also up/down/left/right arrows, and gadgets for moving forward and back vis-a-vis the picture plane, associated XYZ toggles, and alternate coordinate system gadgets. Below all of these is a slide gadget for setting numerical operators associated with many of the gadgets listed (e.g., telling the system how far to move you in or out). There is an absolutely essential "Stop Gadget" that aborts changes in progress.

PR3D's other commands and selections are hidden from sight in the Title Bar's pull-down menus, several of which are examined here:

SETUP MENU

AUTODRAW/AUTOCLEAR (these should be toggled off when working on the screen. It saves time. Then, when done, just hit "Draw" in the toolbox).

Draw modes (all the options you'd ever want: ull, outlined, lines at edges, wireframe, work object enhanced so it can be seen better, and front facets only which saves redraw time. There is also a dither level selector here ... off, 1, 3, and 7).

STEREO (PR3D supports the Haitex stereo glasses and red/blue glasses. There's also a stereo-off setting here).

LIGHT (four settings: OFF, which allows objects to be drawn in their natural color; NATURAL, which shades objects proportional to their distance from a light source; MORE, which gives objects lighted facets inversely proportional to their distance from a light source; STRANGE, which does not prefigure distance from the source in lighting object facets).

WATTAGE (allows you to input the strength of the light source. PR3D could use more than one light source for more varied effects).

DISPLAY (here's where you tell the system what you want the screen resolution to be. PR3D handles all Amiga resolutions. There is also a bitplane requester, offering choices from 1 to 6).

PAGE SIZE (SuperBitmaps can be input, dependent upon the memory available in your system. Also full, half, quarter, and eighth Pages).

PAGE (dimensions and position of the upper left corner in XY coordinates).

DistOPP (this is a PR3D special term meaning "distance from observer to picture plane". The slider below the toolbox determines the numerical distance in centimeters).

PRINTER (allows you to set the number of columns and rows your printout will be).

GENLOCK (toggles transparency of color 0).

DEFAULT (loads the default screen settings as a lo-res non-overscanned 32-color screen, unless you change the default).

LIBRARY

The "primitives" already included here (nine named and twenty-one numbered figures) combined with the "Array" function allowing you to create and animate some really wild and intriguing shapes. There is also an alphabet under "Letters" and numerals under "Numbers". These can be extruded for quick logo production. You can also set the Disk and Drawer path for accessing your own 3D library of shapes.

THE SCRIPT MENU

Scripting is really the heart of controlling PR3D. As you input mouse/icon selections, the 3D>CLI screen (always in view) translates selections into scripted format, so that in no time you are motivated to memorize the scripting commands. Scripting is actually a much faster way to produce results in PR3D, as it contains possibilities for generating images and animations that far outstrip the icon or menu selection methods. PR3D can also be controlled through AREXX commands, which makes scripting all the more essential to learn. In the Script Menu, you can set paths for recording, reading in, editing, and deleting script files.

I certainly did not anticipate the quantity of new tools in my first experience with this software. My head was also turned by the way that PR3D allows dithering of palette colors, giving you a range of apparent colors in hi-res that go beyond the 16 color limitation. It is this capability of PR3D that I would like to dwell upon in depth, because with it, you can create some truly stunning multicolored graphics and animations.

I like to work in hi-res (640 x 400). This gives my visuals the sharpness that I desire, while minimizing the dreaded "jaggies". A severe limitation of hi-res, however, is the Amiga's 16-color hi-res barrier. If your objects are simple, sixteen colors can be enough to support the illusion of 3-D. It may take all of the sixteen to provide a range of darks and lights that provide smooth transitions of light and shadow, but it can be done. If your designs call for complicated multicolored shapes, however, the situation becomes more difficult. Working in the normal Amiga 16 color hi-res palette, you are forced to assign less palette pots (positions) to each color range. This, in turn, severely limits the believability of the finished pseudo-3D shapes. You could choose to work in HAM mode, but there again even the best resolution (320 x 400) isn't fine enough for delicate shapes and diagonal angles without an appearance from the jaggies, plus HAM animations are usually very space intensive.

There have been at least a dozen Amiga articles dedicated to creating more colors in all Amiga resolutions by embedding dot patterns of varying densities into color brushes. By doing this, the Amiga artist can expand the apparent color ranges of an electronic painting. The technique is simple enough, and is comparable to what is called a "screen overlay" in printing. The eye sees the color plus its semi-transparent screen as a new color. Either color screens or black patterns can be used. The idea is old, and its application to Amiga painting has been documented. Only MindWare's PR3D, however, has made this technique an operational part of a working rendering program.

PR3D reserves 28 color registers for facet colors when in lo-res mode (32 colors normally available without dithering). There

are also seven levels of dithering possible between any two colors. Dithering, if not chosen correctly, can look very grainy. Lower resolutions show the dithering very exaggerated as well. You can either hate this, or actually go for it as a specific "look". The color of the dither effects the overall color of the image, so care must be taken in its application. You can program the dither color to change with each animation frame, altering the whole animation with shimmering colors. Now let's take a detailed look at a hi-res example.

THE HI-RES EXAMPLE

Before designing a complex multi-colored shape in hi-res, let's discuss PR3D's lights for a moment. Placement, selection, and "wattage" of a light will effect all forms in the environment. There are three settings for the light source: "Natural", "More", and "Strange". Natural tries to mimic the light we experience in the everyday world. More light intensifies the apparent brightness, and is inversely proportional to the distance from the source. Strange lights are not effected by distance, only by the angle of the source. Setting the wattage of a light can increase or decrease the global brightness of the screen.

Now for a complex and multicolored hi-res object. To begin, I never work on the perspective screen, but prefer a front or side view. This allows me to adjust my objects with a finer understanding of what I'm doing. This object starts by using the cylindrical extruder to create a single shape with fifty faceted sides spun around a central point (Figure 1). Notice the fine degree of shading. Next, I've added a square frame around the top (see Figure 2), and colored it red (I make a square frame by telling the lathing function to give me only four facets). Finally, I have added several other 3-D color bands around my original shape. With each addition, the degree of dithering becomes more obvious to the eye. Just to complicate matters, I'll turn the view to "top" and add two vertical color bands. When complete, I select "Merge All" so that the whole figure can be set to revolve or animate in whatever fashion. I've turned it on an angle so you can appreciate the complexities of the form and the dithering that PR3D provides (Figure 3). The whole figure can now be animated in several ways. When animation is underway, the dithering that is so obvious is superseded by the apparent motion. All you see are washes of color that responds to changes in the light.

Mindware will upgrade PR3D in the near future, adding such niceties as real ray tracing and other options. If you haven't investigated PR3D on your own yet, and you are interested in Amiga graphics (PR3D frames can be saved as IFF files) and animation, do yourself a favor and consider its purchase. I have found that Mindware not only provides quality products, but great service as well. Well, that's all for now. See you in ROMulan space. Enjoy.

•AC•

PageRender3D
Mindware International, Inc.
110 Dunlop Street West
Box 22158
Barrie, Ontario, Canada L4M 5R3
(705) 737-5998
Suggested Retail Price: \$129.95
Inquiry #217



WE ARE AMIGA HEADQUARTERS!



SOFTWARE

Bandit Kings	36.00
Arexx	30.00
Battle Chess	30.00
Can Do	90.00
Champions of Krynn	30.00
Chaos Strikes Back	24.00
Clue	24.00
Colonel's Bequest	36.00
Conquests of Camelot	36.00
Defenders of the Earth	24.00
Deluxe Paint III	90.00
Deluxe Print II	48.00
Double Dragon 2	24.00
Drakkhen	36.00
Dungeon Master	24.00
Dyter 07	24.00
East West Berlin	24.00
F29 Retaliator	30.00
F/18 Interceptor	13.00
F-19 Stealth Fighter	36.00
Falcon	30.00
Ferrari Formula I	13.00
Fighter Bomber	36.00
Fool's Errand	30.00
Ghost Busters II	24.00
Hard Ball 2	30.00
Harmony	27.00
Highway Patrol 2	27.00
Keif the Thief	13.00
Kings Quest IV	36.00
Leisure Suit Larry 3	36.00
M1 Tank Platoon	36.00
Man Hunter 2	30.00
Maniac Mansion	13.00
Math Blaster Plus	30.00
Neuromancer	27.00
NY Warriors	30.00
Operation Combat	30.00
Phasar	54.00
Pirates	27.00
Police Quest 2	36.00
Pro Write 3.0	105.00
Red Storm Rising	33.00
Risk	24.00
Scene Generator	24.00
Shadow of the Beast	24.00
Beast II	36.00
Shark Attack	24.00
Sherman M4	30.00
Sim City	30.00
Sonix	90.00
Swords of Twilight	13.00
Teenage Mutant Turtles	27.00
The Plague	24.00
Their Finest Hour	36.00
Turbo Silver	120.00
TV Sports Basketball	30.00
TV Sports Football	30.00
Viking Child	24.00
Waste Management System	25.00
Wings of Fury	24.00
World Trophy Soccer	30.00
Zak McKracken	13.00

HARDWARE

Amigen	145.00
Baseboard (0-4)MB A500	110.00
CA 880 Disk Drive	124.00
Color Splitter	114.00
Fast Card Plus	210.00
Flicker Fixer	379.00
Flicker Master	11.00
Golden Image Mouse	39.00
Internal Disk Drive	84.00
Mini Gen	203.00
Migraph Hand Scanner	319.00
Mini Max 4	130.00
Optical Mouse	62.00
Perfect Sound	66.00
Pro Gen Genlock	340.00
Supra Drive Ext. 3.5	100.00
Supra Ram 500	59.00
Synchro Express 500	49.00
Video Toaster	1495.00
Vidi-Amiga	132.00
Vo Rec One	107.00
Word Sync Controller	118.00

**SONY DS/DD
100 PACK**

\$61.00

To Order Call
1-708-893-7464

New products arriving daily!

VISA/MASTERCARD/DISCOVER - No Surcharge
Illinois Residents add 6.75% Sales Tax

Shipping \$3.00 per Order. COD Shipping \$3.30 additional.

Defective Software will be replaced with same program
within 30 days. Prices subject to change without notice.

FD Software
P.O. Box 68
Bloomington, IL 60108
Hours M-F 11-7 Sat 10-6

3-D Professional

by David Duberman

IN CASE YOU'VE BEEN WONDERING WHAT'S IN THAT HUGE GREEN BOX YOU'VE noticed recently at the local Amiga retailer, here's the straight skinny. 3-D Professional, developed by Cryogenic Software for Progressive Peripherals & Software of Denver, CO, contains three manuals, six disks, and an instructional videotape.

The tape expertly illustrates most of the tutorial material that makes up the bulk of the user manual. There's also an exhaustive 450-page reference manual and a guide to Animation Station, which is included on two of the disks.

3-D Professional is a three-dimensional modeling, rendering, and animation program. It doesn't perform ray-tracing, but that's promised by the developers for future revisions. The object editor offers a wealth of options and is very well designed, permitting easy, rapid creation and manipulation of a practically unlimited variety of objects in a solid, realistically shaded mode, but no vertex-level creation or reshaping. The rendering engine allows a very wide range of surface choices including varying degrees of transparency and a rich selection of rendering options, and yields great-looking images. As for 3-D Pro's animation capabilities, while it lets you do tweening and morphing easily, you must create text scripts to accomplish path animation.

The program isn't copy-protected and includes a hard disk installation program which works fine, but only copies enough of the example files to let you follow the tutorials. Also included is a version which works only with 68020 and 68030-based Amigas with floating-point math coprocessors, which naturally works faster. While 3-D Pro multitasks, it prefers to have the Amiga all to itself and closes the Workbench screen if possible (you can reopen it if you like). Of course, as with all such processor-intensive software, the less that's going on elsewhere, the quicker you'll get results. 3-D Pro offers an option to toggle the screen display while rendering to speed things up even more. If you leave the screen display on, the program shows a countdown that doesn't seem to have any direct relationship with

how much time is left. And to tell the truth, 3-D Pro is a bit slow at rendering final images, even on a 25 MHz Amiga 3000, although screen redraw in the editor is quite fast.

THE OBJECT EDITOR

3-D Professional is easy to learn, whether or not you've used other 3-D programs. When the program starts you're placed in the object editor's camera view, which shows the scene roughly as it will look when rendered. While the camera view is adjustable interactively, it doesn't permit interactive editing of objects. For that you can switch to any of six fixed full-screen views—front, back, top, bottom, and left and right sides—that allow interactive selection, movement, copying, and linking of objects. There's also a Model view that simultaneously shows the top, right, front, and camera views. Model view also doesn't permit any direct editing, but you can use it to pick an edit view. The editor uses the Amiga's 16-color medium resolution (640 x 200) mode, applying dithering to represent shaded objects. The ability to see solid objects realistically colored and shaded while editing, making adjusting of lighting much easier than otherwise, is one of 3-D Pro's many real strengths. You can also opt for wireframe or solid-color representation, both of which are faster but less realistic, and use the objects' true colors. Similarly, you can abort a screen redraw with a press of the spacebar if you want to make several changes in a row. These and many other small but useful touches help make 3-D Professional a joy to use, for the most part.

Many editor commands are available from a vertical two-column icon-based toolbox at screen right, similar to Deluxe Paint's interface. The top six icons allow rotation of the camera (in camera view

mode) or of selected objects in the directions right, left, up, down, clockwise, and counterclockwise. In both cases, the program uses an ingeniously designed interface for setting the angle of rotation. When you select any of the rotation gadgets, a large circle appears superimposed over the editor screen, with a shaded portion representing the desired angle, which defaults to 90, while the current angle and rotation settings are shown as numbers in the corner of the screen. Just double click on the appropriate icon to rotate by 90 degrees, or move the mouse to adjust the angle with graphic and numeric feedback, clicking the button at the right setting. This scheme would have been perfect if the designer had also permitted keyboard entry of the angle. You can also pan the camera in any direction using a mouse-keyboard combination.

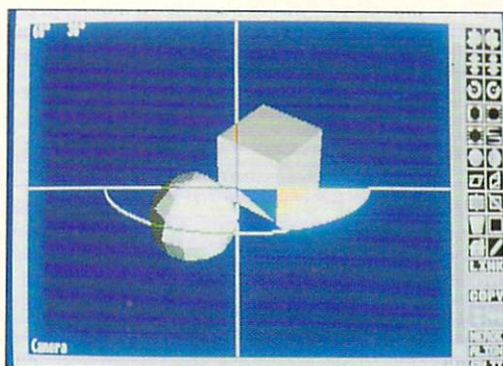
There are icons to scale objects (or zoom the camera view) vertically, horizontally, or both, and optionally to scale objects by the same amount in all three dimensions. In the six edit views you can scale and pan the view using sliders at the window's edge. If you lose your objects by zooming in or panning over too far, an arrow at the window's edge shows you which direction to scroll in order to find them. There are icons to let you flip and shear objects vertically and horizontally, reverse the order of an object's polygons (sometimes necessary for inside-out objects), and to slice objects, which adds the extra faces sometimes needed for realistic shading. If you're familiar with VideoScape 3D, 3-D Pro uses a similar object structure in which polygons are one-sided, can have more than three edges, and ordinarily must have vertices arranged in clockwise order to be visible. This results in faster rendering times if all objects use the proper structure. It is possible to force rendering of all polygons regardless of the direction they're facing, which slows rendering.

Each object has its own axis, a straight line that can be interactively set to any angle and placed anywhere. The arbitrary rotation icon then lets you rotate the object

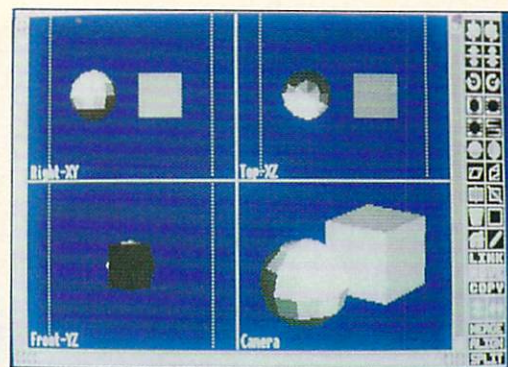
about its own axis rather than the universal axes. The remaining icons let you delete objects, toggle the grid display, return the camera to "home" position, and redraw the display.

Three buttons below the icon tools set Link, Move, or Copy mode. Use Link mode to set up (and view) hierarchical relationships between objects. To link a propellor to its plane, for example, simply drag a line from propellor to plane. In this example, the propellor moves the same way as the plane does, preserving the physical relationship between the two, but can also move (e.g., rotate) independently. Conveniently, links can be temporarily disabled via a menu command. Link also lets you set up light and camera tracking, a unique feature in 3-D Pro in which a spotlight or camera continually adjusts its position during an animation to follow a moving object. And while we're on lighting, 3-D Pro allows a total of five sources, each of which can be the traditional point source or adjustable cone-shaped or cylindrical spotlights. Of course you can adjust each light's color and intensity, and you can set a light's illumination to decrease with distance or not (like the sun), and whether it casts shadows (the latter feature isn't yet supported in 3-D Pro). Lights can be switched on and off independently, which makes it easy to experiment with different effects. You can also set ambient light color and intensity as well as that for an optional haze effect that "fogs out" the background.

Move mode is the default editing mode and allows for easy positioning of objects. Click anywhere on an object and it's surrounded by a white bounding box clearly delineating its extent and showing that it's selected. By the way, you can turn off objects completely and show only their bounding boxes



3-D Pro's ingenious interface device for rotation.



Right, top, front, and camera views shown simultaneously.

(with an identification number to keep track) for extremely fast editing—a very nice touch. Click on other objects and they're selected as well. Click on a selected object to deselect it, or click in a blank area to deselect all selected objects. If you can't select an object that's "surrounded" by a larger one, you can "lock" out the larger temporarily to prevent its being selected. Click and drag on an object or shift-click on any of a selected group and it or they are instantly deposited in a new location. In general, the interface is quite intuitive.

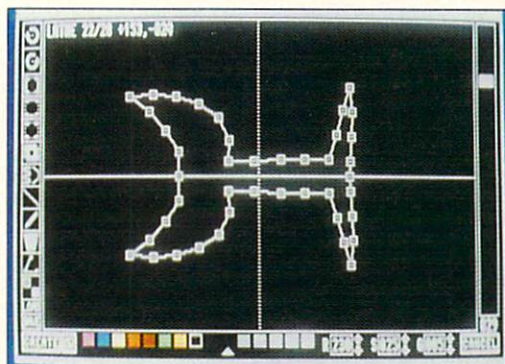
Two gadgets let you lock out vertical and/or horizontal movement for precise alignment. Reposition lights, the camera, and objects' origins (rotation points) just as easily, but you can recenter an object's origin with a menu command. You can even aim the camera interactively by dragging the line of sight, a one-pixel thick line that shows the camera direction but can be difficult to grab. To clone objects, select Copy and then just click and drag as in Move. You can also group, align, and split multiple selected objects.

3-D Professional abounds with friendly touches that are most useful for working artists. If you're editing a complex

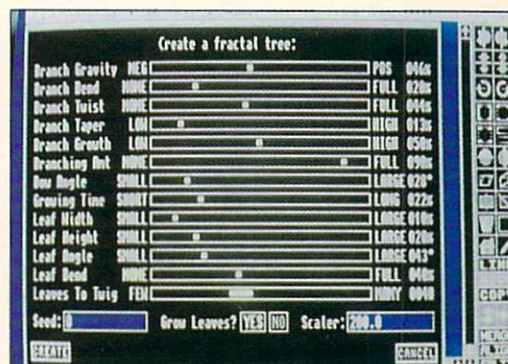
scene whose redraw time is starting to bog down, you can speed things up quite a bit by letting the program turn off the display during recalculation. Or if you're in a hurry to see the final results, you can force redraw to use Extra mode settings (e.g., HAM Interlace, smoothing, etc.), which I'll describe shortly. If moving several objects, you can turn off redraw after moving—only the outline indicates the new position. You can opt for a numeric coordinate display in terms of screen pixels or world units, and toggle the display of lights, the camera, and object origins and axes. And that's not all by a long shot.

OBJECT CREATION

Unlike Sculpt 3D/4D, Turbo Silver, and others, 3-D Professional doesn't let you create three-dimensional objects interactively point-by-point and face-by-face (an arduous process at best). But if you want fast, easy, and versatile creation of a wide variety of objects, then 3-D Professional may well fill the bill. First, the Primitives menu conveniently shows pictures of the built-in shapes: cube, pyramid, three complexities of sphere, tetrahedron, cone, cylinder, wheel, wedge,

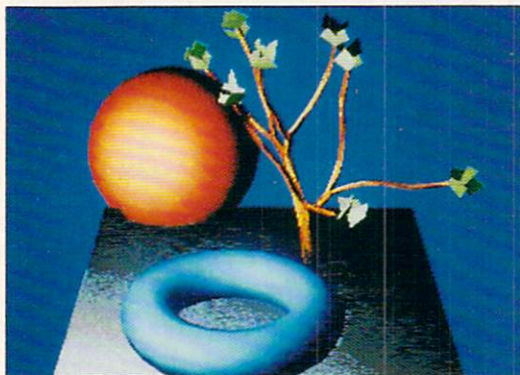


Object outline for spinning via lathe tool.



Object creation tool. Note the many parameters for creating tree objects.

hemisphere, torus, and line segment. While you can stretch and squash these, you can't alter their basic structure. But even more importantly, 3-D Pro can import objects from many other programs: VideoScape3D/Modeler3D, the Sculpt series, Turbo Silver, 3Demon, Forms in Flight, AutoCAD DXF files, and even the Atari ST programs CAD 3D and CyberSculpt. The reference manual gives



top: An example of the beautiful shading capabilities of 3-D Pro; bottom: Landscape created entirely in 3-D Pro and rendered in HAM.

detailed information on importing these formats, but it's generally not necessary because the program automatically recognizes any supported format! Also, 3-D Pro can save any object in its own or in VideoScape format, so it works quite well as an object-conversion program.

Perhaps the most spectacular internal object-creation capability is the fractal tree-making utility. Among the 16 user-settable variables used are branch twist and bend and leaf size, angle, and bend. You can spend many productive hours playing with this part of the program alone. For the most part, the trees produced are natural-looking without being overly complex, an impressive feat. There's also a fractal landscape utility that generates realistically colored rectangular mini-worlds complete

with lakes, plains, and hills—you can vary the size, complexity, and overall roughness. The Ground Creation Tool lets you create a flat ground that extends throughout the 3-D universe, either solid-colored, checkered, or a wireframe grid. There are also utilities for creating three-dimensional objects out of keyboard-entered text in any Amiga font or from imported IFF bitmaps. These are slow and inefficient, creating needlessly complex objects by making a separate polygon for each pixel in the text or bitmap.

The three other object creation tools are Lathe, Profile, and Conic. Lathe uses a special window that lets you create and edit an outline that is then to be spun about a central horizontal axis to create a solid radially symmetric object, much as in actual lathing. You can add points to the outline and delete them, line several consecutive points up in a straight line or create curves. The outline is mirrored on the bottom half of the screen to give you a better idea of the final lathed object's shape. Profile, a more powerful object creation tool, neatly combines the traditional extrusion concept with the "cross-section" technique used by Turbo Silver's Skin and Sculpt's Unslice commands. It lets you easily define complex objects such as boats or human heads by creating any number of parallel slices or cross-sections, then covering them with a continuous skin. Cross-sections must each have the same number of vertices. The Conic tool extrudes a single outline to a point, which seems to be of limited use.

Once you've created or imported an object you'll want to edit it using one or more of the Edit Object requesters, a subject that occupies over 40 continuous pages in the reference manual. In the main requester you set an object's two colors, special rendering flags, surface properties, and texture. An object in 3-D Pro has two colors, or rather two sets of colors. The first, called indirect color and used for coloring objects in the work views, is limited to the editor's 15-color palette, of which you can edit only the first eight colors, the others being a fixed gray scale used for program requesters. The direct color is set via 8-bit RGB sliders resulting in a choice of over 16 million colors, is visible only when rendered using direct color setting, and is accurate only when rendered to 24 bits and displayed on a frame buffer. The only frame buffer directly supported by 3-D Professional in the initial release is

Mimetics' FrameBuffer. The Special Rendering Flags let you force how the object is to be rendered, such as wire frame, solid unshaded, or even invisible, no matter what the global smoothing and shading settings are. One flag lets you render an object as a shadow, invisible itself but darkening any surfaces it covers.

The Surface Property settings include transparency, reflection, roughness, specular reflection, glossiness, and even index of refraction, which isn't currently supported. There are nine preset combinations of these which do a good job of simulating substances such as plastic, ceramic, and stone. And since 3-D Pro doesn't support IFF brush mapping, or the application of arbitrary 2-D images to 3-D objects (as in Turbo Silver), it's fortunate the designers have included a number of well-designed textures for applying to objects. The three texture types are Color textures such as wood, marble, spotty and blend; the Normal textures, ripple and bumpy, which appear to change the object's physical surface; and the Post texture, snowy, which is applied after the others. The textures have lots of user-alterable settings and most look significantly better than those supplied with Turbo Silver SV. However, unlike Silver, you can't apply different versions of the same texture to different objects in the same scene, as noted in the manual. Actually, the textures are "global". For example, if you make several wooden spheres they'll look as though they were carved from a single block of wood and each was retained in the exact spot from which it was carved. If you animate textured objects they appear to be moving through the texture—an unusual effect, and not always a desirable one.

The Object Information requester, available from the attributes requester, displays the object's vital statistics: the number of primitives and vertices; the current setting for rotation, scaling, and shear; the current extents in the X, Y, and Z directions; and the origin and object ID number.

The final object requester, called the Object Show requester, lets you select parts of objects for recoloring and so on. Normally new object settings apply to the entire object, but from here you can deselect all object polygons, then simply select the ones you want to change, or vice versa. You can only select one polygon at a time in a method that's slightly more cumbersome than that used by other Amiga 3-D programs, but you can rotate your

object in any direction during selection without affecting the actual object's position—another nice touch.

RENDERING IMAGES

The Extra menu should more properly have been called the Render or Display menu, except that it leads off with the Settings submenu. The three settings are: whether or not you're warned when memory is low; whether an audible beep is used to signal the user; and whether the program saves icons with files. You can also save and load environment files, which contain a number of other settings including the remaining ones from this menu.

Other than that, this menu strictly controls settings for 3-D Pro's rendering engine, called the Extra Display by the designers, including the Display command itself. From here you also set overscan, interlace, and the graphics mode for rendering: low or high resolution; two versions of HAM mode which differ in the means by which the palette is calculated; 2024 which renders to the high-resolution black-and-white monitor of that model number using 256 shades of gray; and size, which lets you set a custom screen size up to 1024 x 1024 pixels.

The Extra Rendering submenu contains 12 items arranged in several groups. The first group lets you choose the type of shading; pattern or solid, or one of two types of smoothed shading called Gouraud and Phong. Alternatively, you can use a custom-written renderer, an option that will probably be used by the ray-tracing engine when it's available. Other options determine use of the Z buffer for more accurate rendering of objects in front of each other, toggle use of direct colors, and allow IFF images to be used as foregrounds and backgrounds.

There are five different types of dithering, including Random and None, and fully ten different options for output, including Normal (the screen), IFF file, Mimetics' FrameBuffer, and Full or Encapsulated PostScript. And of course there's the Display command, which begins rendering.

Last but not at all least on the Extra menu are two special real-time animation functions: Real-time Rotate which lets you control a rotating camera around the scene; and Simulate which lets you move the camera freely throughout the universe. Both use special keyboard commands for real-time control.

GETTING TECHNICAL

The Tech menu offers a number of complex options that are unique in the world of Amiga 3-D software, at least to my knowledge. Clipping determines the method by which program decides how to eliminate objects and parts of objects outside the display area—the choice is 2-D, 3-D, or none. HS Sorting sets the way points, lines, and polygons are sorted before rendering in order to find out which are behind or in front the others—choices are MinZ, MaxZ, AvgZ, CntP or Center Point, the default, and None. Back Face Removal determines whether polygons facing away from the camera are rendered. With some objects you may have to force rendering of all polygons, or simply force all polygons to face the camera. Other options include double edge removal and exact color on, which forces pixels in HAM pictures that match palette colors (e.g., black) to the palette color, which is required for genlock applications, but slows rendering.

Further Tech menu options include: the transparency buffer, which is necessary when rendering transparent objects using a Z buffer mode; angle smoothing, which smooths angles 90 degrees or greater, such as a cube's edges; curve opacity, which lends greater realism to curved transparent objects; and image filter, used to soften or "smear" an image with a 3 x 3 or 5 x 5 matrix. Finally, since 3-D Pro doesn't normally load fully in order to save memory, accessing overlays or segments from disk as necessary, you can opt to have loaded segments remain in memory or be unloaded when not being used, or you can just load or unload all segments.

3-D PRO ANIMATION

There are two basic methods for creating animations, other than using text-based scripts, which allow you to execute almost any program function remotely. The first is manual, where you make all the changes by hand, then render the frame, then save the frame to an animation file—a laborious procedure to say the least. Most users will prefer the automatic animation method, in which you make major changes in the scene, creating animation nodes that are saved as key frames, then let the computer create the in-between frames necessary for smooth motion. Options offered here include the number of in-

between frames per key and whether motion between key points is straight-line or smoothed into curves.

While 3-D Pro makes the process of creating a key frame list absurdly easy, and you can edit each key frame, you can't edit



top: Work showcasing brick, wood, marble, spotty, and blend textures, as well as three different trees; bottom: Spheres with checked, wood, marble, and spotty textures.

the animation per se. To change an animation's order of sequences, for example, you must recreate it from scratch. Still, the ability to easily perform convincing character-style animation using tweening makes the extra trouble worth it.

The program saves animations in the IFF ANIM opcode 5 format, which is compatible with just about every other Amiga animation software. 3-D Pro can load and display any IFF ANIM file. It can also load and display IFF still images, RGB-format files from various sources, and even X-Specs 3-D pictures. You can also save and load entire scenes with all camera settings, lights, etc. The program recognizes eleven different types of files, including objects, lights, palettes, ANIMs, etc., and uses the appropriate directories that are set up during installation. This lessens the user's burden by not requiring

(continued on page 61)

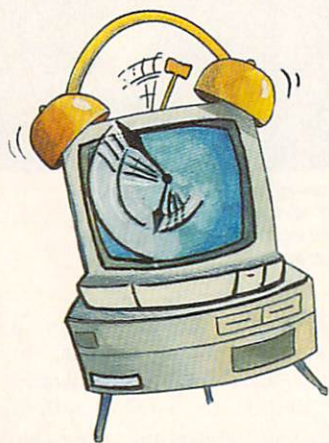
Time Out!

by Mark Cashman

AN AMIGA PROGRAM OFTEN NEEDS TO BE SIGNALLED when a specific amount of running time has elapsed. In some cases, this signal starts up a suspended program; in other cases, it interrupts the normal activities of the program, causing control to be transferred to an event-handling routine.

The Amiga supports time in hardware. A high-resolution clock chip is used for detailed system timing. The video clock (VBlank timer) is used for long-term timing. Other hardware devices, such as the disc, keyboard, and mouse (usually called gameport) are also needed by the system software.

A software device—a special type of program—is used to control a hardware device. On the Amiga system, the software device provides a consistent method for communication with the hardware (henceforth, we will mean “software device” when speaking of a “device”). Contrast the software device with the other executable entities in the Amiga operating environment:



Library: a set of routines, a set of pointers to those routines, or a data area for use by those routines. The routines in a library are only performed as a result of a call from a task or process; they do not “run by themselves”. Examples are the “intuition.library” and the “graphics.library”.

Task: a routine which is run by Exec. Unlike a library, a task does “run by itself”. A task can call on other routines within or without the task object code, or it can call on library routines; further, it can send messages to

other tasks or processes. A task is not able to use files which are a service provided by AmigaDOS. The AmigaDOS process is used in place of the task, when file access is needed.

Device: a task and a library in conjunction. The task queues and dispatches messages (requests) from one or more clients to the device library routines (a regular Amiga library), which then reply to the client. Examples are the “timer.device” and the “trackdisk.device”.

Process: a task that can access the AmigaDOS File system. Your program is a process when you start it from CLI or the Workbench.

The “timer.device” is one of the essential devices that is available immediately after startup. Other devices such as “keyboard.device”, “gameport.device”, “input.device” and so on,

are also included as part of the Kickstart (or ROM (read only memory)) software.

Other devices, such as the “parallel.device”, are not loaded into the system until they are needed. These “disk resident” devices are kept in the “devs:” assignment, which is normally the file “workbench:devs”.

A program needs to follow these steps to use a device:

Create a message port to which the device will reply to requests.
Open the device.
Set-up the request.
Send the request.
When the request is returned, do whatever is necessary.
Close the device.

The steps of setting up to use a device and to process a device reply are not difficult, but they are tedious. For that reason, I have defined a module that provides access to the “timer.device”, just as the “timer.device” software controls the hardware timers.

One of the most powerful features of Modula-2 is its MODULE construct. A MODULE is a container for data, type definitions, and procedures. When a module is written separately from the program that uses it, it is called a library module (no relation to an Amiga library). It consists of two parts—a DEFINITION and an IMPLEMENTATION. The DEFINITION lists (EXPORTS) those features of the IMPLEMENTATION that are available to programs that use the MODULE. The IMPLEMENTATION fully defines all of those things mentioned by the DEFINITION (except for EXPORTED TYPE definitions, which appear only in the DEFINITION). A MODULE is IMPORTED into a client module, and then the names of features of the DEFINITION can be used freely by the IMPORTING MODULE. The following are the MODULES used for the example:

Timer.def	DEFINITION MODULE
Timer.mod	IMPLEMENTATION MODULE
TestTimer.mod	Program module

Some special terms:

Client module: A module that imports a library module.

Calling task: A task that calls on the procedures of a library module.

The following is a definition module. It describes the Timer.HandleTYPE data type and procedures available to operate on that data type. Note the use of the suffix TYPE on the name of a type.

Accessing the Amiga's System Timer Device Via Modula-2

```
DEFINITION MODULE Timer;

IMPORT

  Ports;

TYPE

  HandleTYPE;

VAR

  HandleNIL:
    HandleTYPE;

PROCEDURE Wait(Hours, Minutes, Seconds, MicroSeconds: LONGCARD);

PROCEDURE Opened(TimerReplyMsgPortPtr: Ports.MsgPortPtr): HandleTYPE;

PROCEDURE SendRequest

  (VAR TimerHandle:
    HandleTYPE;
    Hours, Minutes, Seconds, MicroSeconds: LONGCARD);

PROCEDURE AbortRequest(VAR TimerHandle: HandleTYPE);

PROCEDURE TimeIntervalComplete(VAR TimerHandle: HandleTYPE): BOOLEAN;

PROCEDURE ThisIsTheTimerMessage

  (VAR TimerHandle: HandleTYPE; VAR Message: Ports.Message): BOOLEAN;

PROCEDURE Close(VAR TimerHandle: HandleTYPE);

END Timer.
```

What does this tell about how to use the timer? The simplest way to use the timer is to call `Wait()`; the calling task will be suspended until the end of the designated interval. All of the setup and cleanup required to use the timer device is hidden within `Wait()`.

Another way to use the timer is to open it and use it to clock off an interval. First, it must be opened. Last, it must be closed. It can, while opened, be used, by sending a request and checking when the request is replied to the calling task, to measure an interval of time. During that interval, the calling task remains active – in the event of a problem, or some other need, the timer request can be aborted. Another procedure is provided to check whether a particular message is the timer reply messages. This can be used to safely remove the timer reply message (with `Lists.Remove()`) from the message port. This might be necessary if that port is being used to receive several different types of messages.

In the second case, the choice of timer reply message port is left to the calling task. This provides flexibility to the calling task's operation. It may already have a message port being used for interprocess communication, or you may prefer to create one specifically for use with the timer.

For a synchronous operation (the calling task continues operation while waiting for the timer reply), the timer is accessed through a handle. The handle is an opaque data type. That is, the client module is not allowed to depend on or access its structure.

The handle is actually a pointer to some other data structure, but because the client module cannot know the structure to which the handle points, the format of that structure can be changed without requiring changes to the client modules, and this is one of its primary benefits.

The client module is responsible for not making several requests with the same handle. It can, however, open several handles at once and use them separately.

Note the special variable declared in the definition module and set in the initialization section of the implementation module: `HandleNIL`. This variable is used in conjunction with the `HandleTYPE`. It allows the client module to initialize the timer handle to a value against which it can be tested to see if it is opened. This will be needed in the error termination logic of the client module. It is also used in the implementation module to help protect against the use of a closed or not opened handle.

The "how" of this definition module is the implementation module.

```
IMPLEMENTATION MODULE Timer;

IMPORT

  IODevices,
  IODevicesUtil,
  Lists,
  MessageUtil,
  Ports,
  PortsUtil,
  TimerDevice;

FROM SYSTEM IMPORT ADR, TSIZE;
FROM Termination IMPORT Assert;

TYPE

  HandleTYPE =
    TimerDevice.timerequestPtr;

PROCEDURE Wait(Hours, Minutes, Seconds, MicroSeconds: LONGCARD);

VAR

  TimerHandle:
    HandleTYPE;

  TimerReplyMsgPortPtr:
    Ports.MsgPortPtr;

  TimerReplyMsgPortPtr:
    Ports.MsgPortPtr;

BEGIN

  TimerReplyMsgPortPtr := PortsUtil.CreatePort(NIL, 0);
  TimerHandle := Opened(TimerReplyMsgPortPtr);

  SendRequest(TimerHandle, Hours, Minutes, Seconds, MicroSeconds);

  TimerReplyMsgPortPtr := MessageUtil.WaitMsg(TimerReplyMsgPortPtr^);

  PortsUtil.DeletePort(TimerReplyMsgPortPtr^);
  Close(TimerHandle);
```


AutoPrompt



by
DigiSoft

**TV SCRIPT PROMPTER
FOR THE AMIGA**

AutoPrompt is a sophisticated scrolling prompter and text edit program designed for TV script prompting and information displays. It has been developed in consultation with TV industry professionals, with flexibility, simplicity and speed as the primary design considerations.

- AutoPrompt has a full intuition interface with menus and keyboard shortcuts.
- Selectable font, font size, pallet, speed and scroll direction.
- Markers for instant movement to pre-defined cue points.
- Message window and graphic scroll speed indicator.
- Simple easy-to-use menu driven text editor with load, save and print options.
- Imports ASCII or IFF FTEXT format text files.
- Supports PAL and NTSC in both interlace and non-interlace, high or low resolution.
- Requires 512k of memory and Kickstart 1.2 or later.

AutoPrompt US \$295.00
Demonstration disk ...US \$15.00

DigiSoft
12 Dinmore St
Moorooka
Brisbane 4105
Queensland Australia

International toll free order numbers from the USA and Canada:

from USA	1-800-525-2167
from Canada	1-800-663-3940
other countries	61-7-277-3255
within Australia	(07) 277-3255
FAX	61-7-277-8473

AutoPrompt is copyright (c) DigiSoft 1990
Kickstart is a trademark of Commodore Amiga Inc.

Circle 137 on Reader Service card.

```
END Wait;

PROCEDURE Opened
  (TimerReplyPortPtr: Ports.MsgPortPtr):
    HandleTYPE;

VAR
  Error:
    LONGINT;

  TimerHandle:
    HandleTYPE;

BEGIN
  TimerHandle:=
    IODevicesUtil.CreateExtIO
    (TimerReplyPortPtr,
     TSIZE(TimerDevice.timerequest));

  Assert(TimerHandle # NIL, "Timer", "Alloc and Init", "Allocated
handle");

  Error:=
    IODevices.OpenDevice
    (ADR(TimerDevice.TimerName),
     TimerDevice.UnitVBlank,
     TimerHandle,
     0D);

  Assert(Error = 0D, "Timer", "OpenDevice", "Opened device");

  RETURN TimerHandle;

END Opened;

PROCEDURE SendRequest
  (VAR TimerHandle:
    HandleTYPE;
```

```
Hours, Minutes, Seconds, MicroSeconds: LONGCARD);

CONST
  SecondsPerHour =
    3600D;

  SecondsPerMinute =
    60D;

BEGIN
  Assert
    (TimerHandle # HandleNIL,
     "Timer",
     "SendRequest",
     "Open handle in use.");

  TimerHandle^.trtime.tvsecs:=
    Hours*SecondsPerHour + Minutes*SecondsPerMinute + Seconds;

  TimerHandle^.trtime.tvmicro:= MicroSeconds;
  TimerHandle^.trnode.ioCommand:= TimerDevice.TRAddRequest;

  IODevices.SendIO(TimerHandle);

END SendRequest;

PROCEDURE AbortRequest(VAR TimerHandle: HandleTYPE);

VAR
  IgnoreReturnedLONGINT:
    LONGINT;

BEGIN
  Assert
    (TimerHandle # HandleNIL,
     "Timer",
     "AbortRequest",
     "Open handle in use.");

  IgnoreReturnedLONGINT:= IODevices.AbortIO(TimerHandle);
  Lists.Remove(TimerHandle^.trnode.ioMessage.mnNode);

END AbortRequest;

PROCEDURE TimeIntervalComplete(VAR TimerHandle: HandleTYPE): BOOLEAN;
BEGIN
  Assert
    (TimerHandle # HandleNIL,
     "Timer",
     "TimeIntervalComplete",
     "Open handle in use.");

  IF IODevices.CheckIO(TimerHandle) # NIL THEN
    Lists.Remove(TimerHandle^.trnode.ioMessage.mnNode);
    RETURN TRUE;

  ELSE
    RETURN FALSE;

  END;

END TimeIntervalComplete;

PROCEDURE ThisIsTheTimerMessage
  (VAR TimerHandle: HandleTYPE; VAR Message: Ports.Message): BOOLEAN;
BEGIN
  Assert
    (TimerHandle # HandleNIL,
     "Timer",
     "ThisIsTheTimerMessage",
     "Open handle in use.");

  RETURN (* TRUE IF *) TimerHandle = ADR(Message);

END ThisIsTheTimerMessage;

PROCEDURE Close(VAR TimerHandle: HandleTYPE);
BEGIN
  Assert
    (TimerHandle # HandleNIL,
     "Timer",
```



```

"Close",
"Open handle in use.");

IODevices.CloseDevice(TimerHandle);
IODevicesUtil.DeleteExtIO(TimerHandle);

TimerHandle:= HandleNIL;

END Close;

BEGIN

HandleNIL:= NIL;

END Timer.

```

As with any real Modula-2 software project, the timer implementation depends on a variety of other modules. Some of these modules are library modules that are delivered with the Benchmark Modula-2 compiler used for this project. Others are general purpose modules which I have created to help with the process of constructing reliable programs. In the case of the above module, the modules which it uses that I have created are MessageUtil and Termination. MessageUtil simplifies some aspects of message passing. For instance, its WaitMsg procedure compresses the usual:

```

MessagePtr:= Ports.WaitPort(MessagePort);
MessagePtr:= Ports.GetMsg(MessagePort);

```

into

```

MessagePtr:= MessageUtil.WaitMsg(MessagePort);

```

The Termination module is used to allow separately compiled modules to properly free up resources on task termination due to normal circumstance or error. The Assert procedure checks the condition of its first parameter, and if the value of that expression is FALSE, then the calling task is terminated with a message. Before termination of the task, a list of procedures registered by various library modules and the task body are called. Generally, these procedures have the responsibility for freeing the resources dynamically allocated by the module from which they were registered.

The Termination module is only used to handle the error case in this Timer module. The TestTimer main module will also use Termination to handle the normal termination of the program.

Thanks to the modularity of Modula-2 programs, it is not necessary to look further into these modules. Instead, turn your attention to the implementation of Timer, above.

The OpenedO procedure returns a HandleTYPE pointer. In the current implementation, this is merely a pointer to a conventional Amiga timer device timer request block. However, a future implementation might add other components to the handle, such as a flag to indicate whether the handle timer request is currently sent out on a request to the timer device. Such a flag could allow the programmer to worry less about whether a program might attempt to send out a timer request more than once with the same handle before the timer device has replied to an earlier request, since it would allow the Timer module procedures to trap and respond appropriately to such inappropriate behavior. All programs using the Timer module would benefit from such a

Professional Quality Full Color Video Digitizing Pro-ResTM Still Video Interface

- * Interface any color video still frame to your digitizer without a color wheel.
- * Control and Modify the Video before you digitize.
- * Preview the Adjustments real-time on your RGB Monitor.
- * Professional Results.
 - Accurate Regenerated Vertical and Horizontal Sync Pulses
 - True 75 ohm Input / output loads
 - Special Sync circuits for Still-video and VCR Video
- * Easy to Use
 - Automatic operation with NEWTEK's Digi-ViewTM
 - Manual Stepping with other Slow-Scan Digitizers

NOW SHIPPING.....RETAIL \$399.--

Full Color PostscriptTM Printing For Under \$5,000.- ?
Impossible....Until Now! Introducing

ColorDriveTM Color PostscriptTM Printer Driver

- * Drives the most common color printers used with the Amiga
 - Up to 300 DPI, 3 & 4 color separations, A & B page size
- * Full Color Desktop Publishing Support
 - Bit-mapped & structured color graphics; color text
- * Downloadable Fonts.....Standard 35 fonts available

DEALER INQUIRIES INVITED

Digi-View is a trademark of NEWTEK, Inc.
Amiga is a trademark of CBM, Inc.
Postscript is a trademark of Adobe Systems, Inc.
Pro-Res and ColorDrive are trademarks of TDA

TDA The Desktop Advantage
560 NE F Street, Suite 4
Grants Pass, OR 97526
(503)476-8254

Circle 110 on Reader Service card.

change, but none would have to be changed to take advantage of it. The client modules would only need to be relinked with the modified Timer module to use such improvements.

The OpenedO procedure checks that the handle object has been allocated, and if it has not, the program is aborted. This prevents the client module from having to deal with such an error which we hope will be rare.

Then the handle object is initialized via OpenDeviceO. If the device cannot be opened, the program is terminated. Note that the Timer module uses the VBlank timer, which operates in increments of 1/60th of a second. A later version of this module might use the higher resolution timer. Again, such a change would be invisible to the client modules, due to the nature of modules.

Then the handle can be closed using CloseO; at that point, all resources are returned to the system. Note that the message port for replies from the timer device is not deleted. This is the responsibility of the calling module.

Also note that there is an attempt at protection, everywhere in this implementation, against using an uninitialized or closed handle. For now, Timer depends on the programmer to ensure that an unopened handle is initialized to HandleNIL. A more sophisticated module might take care of this problem in one fashion or another, although such a solution is not easy.

Note how WaitO is built on the OpenO and SendRequestO procedures. Since it uses a private unnamed message port, WaitO just uses WaitMsgO to determine when the time interval is

NOW F-BASIC 3.0 SHIPPING

The reviewers have labeled **F-BASIC**:

The **FASTEST** Growing
FASTEST Performing
AMIGA Language

Beats C And Other
Basics!

NEW-Integrated Editor
NEW-200881 Support
NEW-Even Faster

VERSION 3.0 F-BASIC

*A Beginner
Can Immediately Use
F-BASIC

*An Expert Can **NEVER** Outgrow **F-BASIC**

F-BASIC 3.0 With User's Manual & Sample Programs Disk—Only \$99.95
F-BASIC 3.0 With Complete Source Level Debugger—Only \$159.95

F-BASIC is Available Only From:

DELPHINOETIC SYSTEMS, INC.

Post Office Box 7722
Rapid City, SD 57709-7722

Send Check or Money Order, or Write For Info
Credit Card of C.O.D., Call (605) 348-0791

F-BASIC is a registered trademark of DNS, Inc.
AMIGA is a registered trademark of Commodore/AMIGA, Inc.

Circle 117 on Reader Service card.

complete, rather than using the `TimeIntervalComplete()` and the `ThisIsTheTimerMessage()` procedures.

The other procedures are trivial. They mostly use the standard Amiga device communication routines, but their inclusion in this module takes a load, particularly that of error checking, off the logic of the client module.

And now, a test program, to verify that this library module works:

```
MODULE TestTimer;

IMPORT

    AmigaDOSProcess
    InOut,
    Tasks,
    Timer,
    Termination;

FROM Termination IMPORT NormalTermination;

CONST

    Self = NIL;

VAR

    MessagePtr:
        Ports.MessagePtr;

    ProcessPtr:
        AmigaDOSProcess.ProcessPtr;

    TaskPtr:
        Tasks.TaskPtr;

    TimerHandle:
        Timer.HandleType;
```

```
TimerReplyPortPtr:
    Ports.MsgPortPtr;

PROCEDURE Report (Message: ARRAY OF CHAR);
BEGIN

    InOut.WriteString (Message); InOut.WriteLine;

END Report;

PROCEDURE Terminate;
BEGIN

    IF TimerHandle # Timer.HandleNIL THEN Timer.Close (TimerHandle); END;

END Terminate;

BEGIN

    TimerHandle:= Timer.HandleNIL;

    Termination.RegisterProcedure (Terminate);

    TaskPtr:= Tasks.FindTask (Self);
    ProcessPtr:= AmigaDOSProcess.ProcessPtr (TaskPtr^.tcUserData);
    TimerReplyPortPtr:= Ports.MsgPortPtr (ADR (ProcessPtr^.prMsgPort));

    TimerHandle:= Timer.Opened (TimerReplyPortPtr);
    Report ("Timer opened.");

    Timer.SendRequest (TimerHandle, 0, 0, 20, 0);
    Report ("Timer request sent. Loop begins.");

    WHILE NOT
        Timer.TimeIntervalComplete (TimerHandle, TimerReplyPortPtr) DO
    END;

    Report ("Loop completed.");
    Timer.Close (TimerHandle);

    Report ("Testing 10 second wait");

    Timer.Wait (0, 0, 10, 0);
    Report ("Back from wait");

    NormalTermination;

END TestTimer.
```

Note the termination procedure `Terminate`. This procedure checks to see if the `TimerHandle` is `Timer.HandleNIL`, and if it is not, the `TimerHandle` is assumed to be opened, and it is closed as part of the termination ritual.

`Terminate` is called by `NormalTermination`, part of the `Termination` module from which `Assert` is imported. Like `Assert`, it calls all of the registered termination procedures to free resources allocated by each module.

Note that this test program does not verify that the request can be aborted. It does, however, verify the major features of the `Timer` module. If the capabilities of the module are increased, tests of the new capabilities can be added.

CONCLUSION

Modula-2's library modules make it easy to codify the handling of devices and other complex hardware and software resources. In this case, timer handling and dealing with errors in the process of communicating with the timer device are hidden in the implementation of the module. The implementation is available to the programmer for improvement without change to the client modules.

Some of the choices of what services to offer, and in what way to present them are arbitrary. They will depend on the envisioned applications of the library module. Some of those choices may prove to be wrong and may need to be changed as the module evolves. Of course, no programmer can anticipate everything. At least Modula-2 makes these changes fairly easy to realize.

•AC•



by R. Bradley Andrews

AN INTERESTING DEVELOPMENT IN THE Amiga games field is noted here in regard to two new European imports from Electronic Arts. In the past they have often taken much time and effort to repackage games in "U.S." boxes with new manuals. This time, they decided to keep the original European packaging. It may not seem like a benefit at all, but in avoiding the delays and costs often associated with bringing European-born games over here, we should find more good games on our shelves sooner. And *that's* the good news!

TURBO OUTRUN

First this month is Turbo Outrun, a sequel to Sega's original Out Run. Both games have enjoyed success as coin-op games. In the Amiga version of Turbo Outrun, you control a high-performance Porsche 959 in a race from New York to California. This is basically a race against the clock, with a few twists. Every fourth city you pass through features a special pit stop where you can add one of three options to your car: high-grip tires, a special turbo, and a high-speed engine. Each is useful for a specific part of the race, and can give you that extra edge you need to come out on top.

The graphics are drawn well, though they fall short of the arcade version. Along the route, the scenery changes based on your current locale, as well as the current weather conditions. The sound is standard—the kind you would expect to hear in a racing game.

Turbo Outrun does have some of the same control problems found in other racing games. However, if you typically enjoy racing games, you should like this one.

HEAT WAVE

Heat Wave, from Accolade, is a different sort of game. It takes you from land to water and puts you in charge of one of four of the most powerful racing boats in use today, in both fresh and salt water courses

throughout the US. The focus in Heat Wave is water speed. Of course, the first one through the course wins. But the game is not that simple. Just as in car races, parts break down and must be fixed during the races. But the only pit stop is at the *start* of the race, and your boat must be loaded with all the supplies it will need *during* the race. Since space and weight is limited on-board, care must be taken to pick the best load for the race at hand.

The graphics are comparable to many flight simulators and use filled polygons for all shapes on the screen. While fairly simple, they do have a certain flair to them, and they are pleasing to look at. The sound is up to the task and helps get players into the feel of power boat racing.

During a race, the screen is filled with dials and knobs that provide feedback on the status of your vehicle. But the largest single portion of the screen is taken up with the main view display. As you might expect, this can be either a view over the bow of the vessel, or from a theoretical chase helicopter above and behind the boat, as is common with most flight simulators.

But all is not rosy. As with many racing games, the ship is very hard to control. Small movements of the joystick cause wide variations in the boat's course. Even using the keyboard does not help that much, since each turn step represents a fairly large change in direction.

These problems make it very hard to successfully complete a course. Not only did I continually find myself going the exact opposite direction of my intended course due to steering difficulties; nearly every attempt ended with my boat crashing into the shore and exploding. The trailing view does allow for a

little better judgement of the exact location of the shore, but proper navigation is still too difficult to achieve for my tastes. An overhead map is available at any time during play, but you cannot control the ship from here.

The designers spent too much time on making the explosions clever and not enough on making the boat easy to control. Avoid this one unless you enjoy watching ships explode.

HARDBALL II

Hardball II continues another series by Accolade. It combines elements of action and strategy to make for an interesting baseball game.



top: Turbo Outrun from SEGA
bottom: Accolade's Heat Wave

Hardball II does a good job of covering all the various elements of baseball. You have direct control over the actions of each of the players on your team. While this control usually focuses on either the pitcher or the catcher, a complete game finds the player controlling all nine positions. Either the keyboard or a joystick can be used for control, but the keyboard is somewhat more flexible. Unlike Heat Wave, the controls actually work fairly well, and true mastery of the game is possible. Given the at times complex nature of the game of baseball, and the wide range of possible actions, it will take some practice.

The graphics in Hardball II are very nice. Three different views are available during play. Overhead provides the least amount of detail, but shows the entire

playing field at once. The views from the pitcher and the hitter show nicely rendered three-dimensional views from each of the respective positions. The graphics are clear and attractively drawn, at a fairly quick speed.

Several teams are included on the disk for use during games. An editor allows the creation and editing of your own teams, so you can input any club directly from the sports page, or from a baseball yearbook. (Accolade did this, rather than include actual teams, to avoid paying the licensing fees Major League Baseball and the Player's Association demand for the use of real team and player names.) Games can also be linked together to form an entire season.

I once had a roommate who loved MicroLeague Baseball II because of its statistical detail, and I am sure he would love this game as well. If you are a fan of the great American summer pastime, this game is sure to provide you with many hours of year-round entertainment.

D.R.A.G.O.N. FORCE

Next is another recent release for the Amiga from Interstel. In D.R.A.G.O.N. Force, the nations of the world have finally tired of terrorism and have put together an elite team of counterterrorists to instantly strike against this menace wherever it arises.

You act as controller of this powerful force. Seven of the fourteen members of the team can be used on each of the twelve included missions. While the goals of different missions might be better served by using different team members, you will likely use the same core group on each mission. Good performance during a mission, along with a lot of contact with the enemy, will improve your group members' statistics, and make them even better the next time.

Once you pick seven team members to send on the mission, you must then outfit them with equipment. The equipment is free, but since each team member can only carry so much weight, you must balance the armament capabilities of your group. Once you have chosen your equipment, the action begins. Each scenario begins after the team has been brought to the drop zone. They must carry out the mission, usually eliminating opponents or destroying buildings, and then make their way to the pickup zone before time runs

out. The action itself is broken into a series of action phases. The player decides the actions each member will take throughout the turn, and then movement and fire is acted out. This process is repeated until time runs out, or the team leaves on the pickup helicopter.

The graphics are OK. The close-up graphics use the pseudo-three-dimensional look I personally like, but Interstel could probably have done better. Sounds are used when shots are fired, demo charges are blown, tanks move, etc. These are well done and fit in nicely with the game background.

The game does lack real punch. The twelve included missions provide good initial entertainment, but once they are done, there is nothing more to do. The game could really have used a scenario editor. And character advancement is very minimal; playing through all the scenarios only develops each character slightly as compared to what one might expect their eventual fighting potential to be. The phased movement also gets in the way of play. It is really tedious to have to move all the forces one by one, each phase. While a move-to command is included to allow a movement command to carry over between several phases, it is impossible to enter more than five steps of an elaborate move, requiring you to input new orders each phase if you wish to do anything except move from point A to point B.

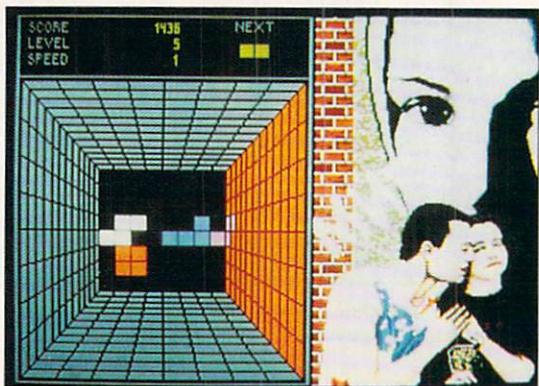
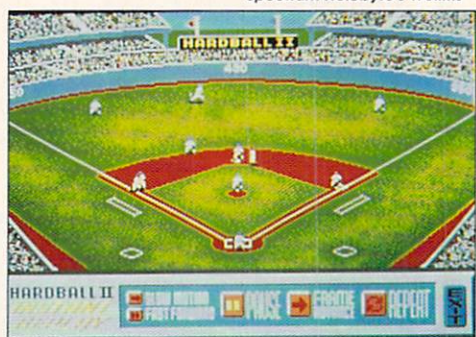
The first game to look at the same subject—man-to-man combat—with a smoother interface and ideally free-flowing action will get my vote. Unless you are really into tactical games, avoid this one.

WELLTRIS

The Soviet Union is getting lots of positive press nowadays, and its pervasiveness in the media has been matched in the game field. Nearly everyone has heard of Tetris, especially since it has been translated to the Nintendo. Not content to settle for just one hit, Spectrum Holobyte has teamed up with designer Alexey Pajitnov to bring a similar game to the US.

Welltris is kind of like three-dimensional Tetris. But instead of dropping pieces down a three-dimensional well as one Tetris-like competitor does, Welltris drops the pieces down one of the four sides of a square-shaped well. Once they hit the bottom, the pieces slide until they are stopped by a previously dropped square, or they hit the opposite wall. The pieces can be rotated as in the original Tetris, but

top to bottom: Accolade's Hardball II, D.R.A.G.O.N. Force from Interstel, Spectrum Holobyte's Welltris



they can also be moved around from wall to wall, provided you have time before they hit the bottom.

When a horizontal or vertical row is filled, it is removed, and all the pieces to one side slide toward the center of the bottom area. The goal is to get all the pieces, in one form or another, completely into the bottom area of the well. Whenever part of a falling piece remains on the side wall after it has stopped falling, that wall is marked out and cannot be used for a short period of time. The game ends when all four walls are blocked out. The game's graphics are rather simple but adequate for play. And just as in Tetris, each play level features its own detailed image of life in the Soviet Union. Sound is simple, too, but works well, and (of course) the pace quickens as you go up in levels.

While I found the game interesting for a bit, it didn't keep my attention for hours on end, as did Tetris. Try it out in the store first to see whether or not it is a "must have" for your game collection.

WATERLOO

Waterloo is a wargame like no other. Originally developed by PSS in Europe, it has been imported by SSI to add to their well-known line of wargames.

As nearly every student knows, Waterloo was Napoleon's final battle, his military genius finally done in by the Duke of Wellington. Not only does the game cover a subject virtually ignored till now, it does so in a truly unique fashion.

Most wargames display an overhead view of the battlefield, as players move around each of the forces on their side with complete control. But not Waterloo. Its views are those seen through the eyes of the commanders (Napoleon or Wellington), from each of their actual locations on the battlefield.

Instead of directly controlling his units, each of the opposing commanders must send orders to his subordinates via messengers. Once an order is decided on, it takes a little bit of time before a courier can deliver the order, and for it to be carried out in battle. Even then, the commander receiving the order will act on it in his own way, possibly even ignoring it altogether, if local conditions lead him to do so.

The game covers just this one battle, from sunrise to sunset on that fateful day. While a day may seem like a short period of time, the game is carried out in fifteen minute intervals, so an entire game does last a long time. Entering the orders can be

a bit of a problem. Only eight orders can be input in each segment, a minor annoyance, but not too much of a problem. The really annoying part is that the orders themselves must be entered in a structured English form similar to that used by most text-adventure games. Yes, it helps if you know your commanders, as well as the landscape, by name. However, I have never really liked text parsers. Some form of reverse parser, using icons or menus for each of the available options at each stage of order construction, would have been much easier to use.

Each three-dimensional view does take a while to draw, but it is crisply rendered and gives a good feel for warfare in the Napoleonic Era. Three-dimensional boxes represent groups of soldiers on the battlefield; they are reminiscent of the imagery used by many tabletop wargames.

Overall, I found this game's graphics and unique approach refreshing, and you should definitely check it out if you are into wargames. Unfortunately, the failure of the designers to include a more simplified interface to control action will probably keep it from most non-wargame players.

PIRATES

I have saved the best for last this month. Microprose has just released their popular game *Pirates* for the Amiga.

Players begin as young ship captains in their quests for fame and fortune during the "Golden Age of Piracy." Each player starts the game working for one of four major powers, having free reign to loot and pillage the forces of whatever countries his host is currently at war with. *Pirates* has both the tactical and strategic elements found in nearly all Microprose offerings.

The main goal is to accumulate personal wealth, including gold, lands, and even a wife of high standing. To do this, a player must successfully captain his crew through a series of missions. All game action takes place in the Caribbean. Players begin in port at one of the colonies of their chosen country, but they have nearly complete freedom in deciding where to go from there. Visits to the governor will update the player on which countries are



top: SSI's Waterloo
bottom: Pirates from MICROPROSE

currently fair game. Governors also occasionally ask personal favors, such as delivering a letter to an undercover relative in an opponent's port. These may not bring in much gold, but they do earn the respect of the Governor, and often bring land grants and increases in stature to the successful captain. Booty can be acquired in several ways; it is then sold to merchants, with the proceeds divided up among crew members. The captain gets a lion's share.

Any ship encountered on the high seas that is defeated and successfully boarded will turn over all its cargo. Captured ships can either be added to the captain's fleet or sunk, depending on the size of the fleet and its cargo storage needs. Enemy ports are like super-rich ships, and can prove quite lucrative to any captain that makes it past the defending forts to defeat any such garrison.

The keyboard, the mouse, and the joystick can all be used for control during play. I found that the mouse works best most of the time, though the joystick was most useful during the dueling segment. Several tactical action segments are repeated throughout play.

(continued on page 95)

Voice- Controlled Joystick

by John Iovine

A
LOW-BUDGET
EXCURSION
INTO THE
WORLD OF
VOICE
RECOGNITION.

RADIO SHACK HAS RELEASED A SPEECH RECOGNITION CHIP, THE VCP200. ALTHOUGH the chip has a limited vocabulary, it is sufficient for a joystick controller.

Voice recognition is a trendy topic. Computer scientists have worked on various algorithms for years. Neural networks recently stepped into the fray, also, with limited success. Some approaches to voice recognition are more successful than others. We will examine one method, the speaker-independent.

SPEAKER-INDEPENDENT

The VCP200 is a speaker-independent voice recognition integrated circuit. Speaker-independent means that regardless of who is speaking to the chip, the chip will recognize its commands. This is difficult to program since everyone doesn't pronounce words exactly the same. Fortunately for us we don't have to do any programming. The VCP200 is already programmed to recognize a number of different commands. There is a trade off for this convenience, most important is a limited vocabulary that we cannot change. In addition this chip can be easily fooled by recognizing non-command words for commands. This disadvantage, however, can be utilized to your advantage. I'll go into this a little later. Considered our voice recognition joystick a low-budget excursion into the world of voice recognition.

SPEAKER-DEPENDENT

Speaker-dependent, in contrast, requires the user to train the computer or voice recognition circuit to recognize the user's voice and commands. This is a more sophisticated approach, and with that lies some significant advantages. First, the commands are usually programmed by the user. Second, the command vocabulary is possibly much larger. Finally, speech recognition is fairly accurate for the user. I plan to design a speaker-dependent system in the future.

VCP200

The VCP200 has two recognition modes: the Command Mode and the Yes-No/On-Off mode. The Mode of the chip is determined by the voltage on pin 19 of the VCP200. By bringing pin 19 low, the Command Mode is enabled. Bringing pin 19 high enables the Yes-No/On-Off mode.

We will be using the Command mode. (See table for command summary and chip pin-out Fig. 1.)

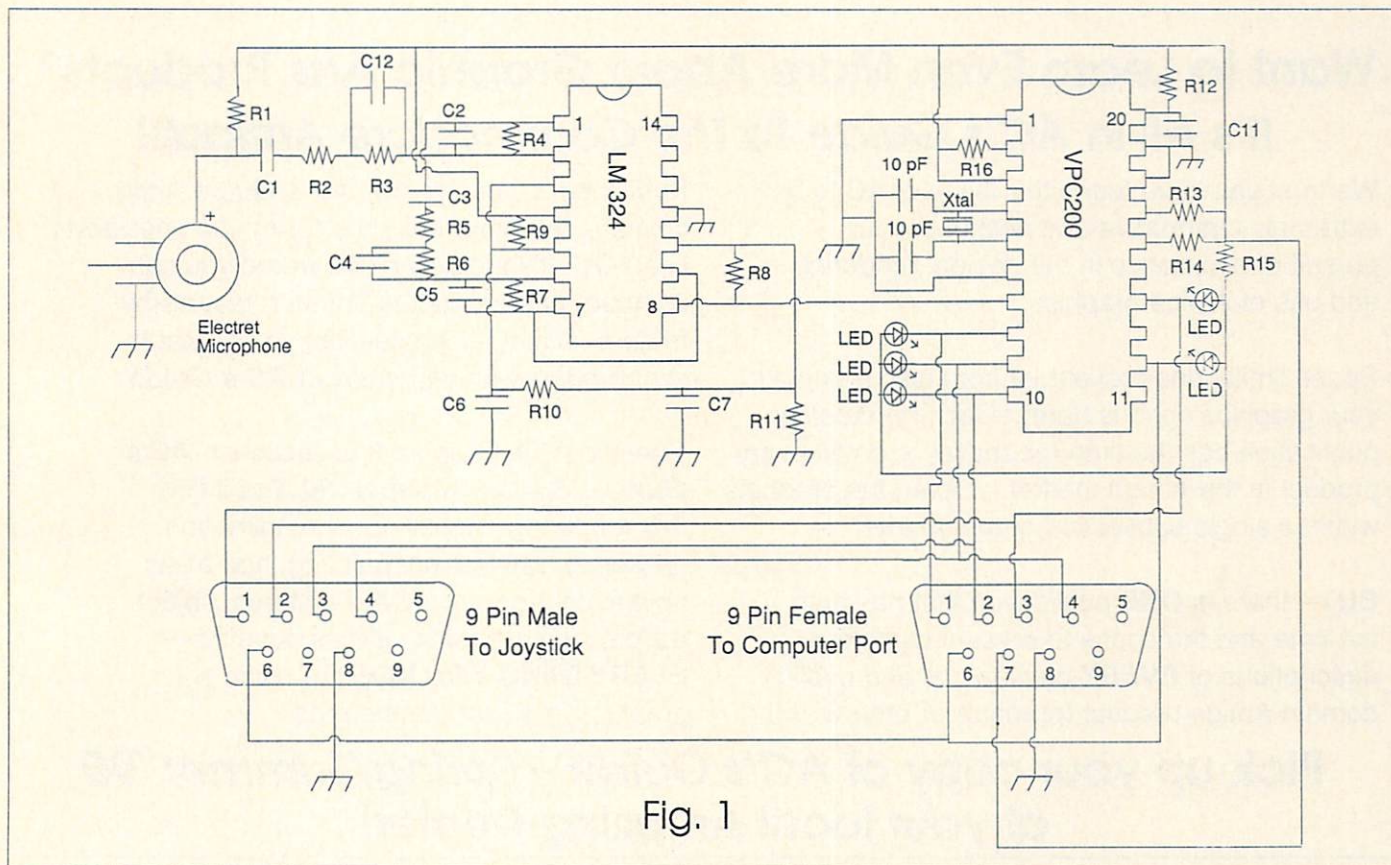


Fig. 1

Joystick Pin #	Joystick Movement	VCP200 Cmd. Mode	VCP200 Yes-No/On-Off Mode
1	Up	Go	Yes or On
2	Down	Reverse	No or Off
3	Left	Left Turn	
4	Right	Turn Right	
6	Fire	Stop	

CHIP OPERATION

The literature that comes with the chip describes the VCP200's basic recognition operation. The chip performs a spectral analysis of the incoming audio signal from 300 Hz thru 5500 Hz. From this analysis it determines the phoneme classes and stores it in a string. Then it compares this phoneme string with phoneme strings it has stored on board. When it finds a match (recognition) it enables that control pin. This is an interesting feat because this is all happening in real time.

My guess is that this chip is using a circulating serial register on the input, although that kind of information hasn't been given.

CIRCUIT OPERATION

The circuit (see Fig. 2) is very similar to the user schematic that comes with the chip. I made minor changes to some component values. Although these changes are minor there are significant when interfacing into the computer joystick port. It appears that the computer generates sufficient RF to jam the circuit. By adding cap. c-?? 220 pf we can minimize this interference and obtain reliable operation.

In addition I changed the LED's to subminiatures and increased the resistance of the current limiting resistor to minimize

the current draw on the port. Remember you can only draw 100 ma. max per joystick-port. The LED's are not essential for proper circuit operation. I left them in for visual indication. You will find the visual indication very helpful when you first start using the circuit.

An on-off switch is essential, this is tied into the +5 volt line. Without this switch you may encounter keyboard problems. Keep the switch in an off position when powering up the computer and for all normal operations. To use the circuit first load your test program or game and start it running before you turn on the circuit. Afterwards when you're ready to quit, turn off the circuit before you end the program. Normal joystick operation is available with the circuit on or off. When it is on however you have visual indication via the LED's of the relative position of the joystick.

TRY-OUT

When you have completed the circuit and have it installed, you do not need to load a joystick program to test the circuit. You can use the LED indicator lights. The LED connected to the command pin (see

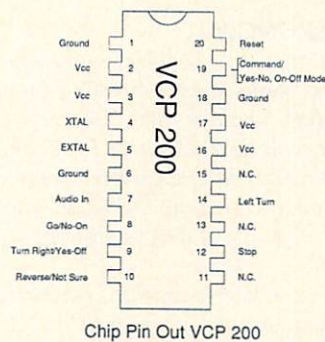


FIG. 2

Want to Learn Even More About Graphic Arts Products?

It's All in AC's Guide To The Commodore Amiga!!!

We trust you have found this issue of AC to be extremely informative and helpful in your pursuit of excellence in the design, rendering and use of Amiga graphics.

Space limitations prevent us from discussing all your graphics options here; in fact, no *monthly* publication has the time (or space) to detail every product in the Amiga market, or even the products within a single subsection of that market.

BUT—there is **ONE** publication that has both the time and the space to present complete descriptions of **EVERY** commercial and public domain Amiga product (graphics or otherwise)!

Published 3 times yearly, **AC's Guide** is the ultimate one-stop reference for Amiga specialists like **YOU!** It's the only place where you can compare one product to another, feature-by-feature, at just a glance—but you'll want to spend *hours* with your copy of **AC's Guide!**

Check out the Graphic Arts section in **AC's Guide – Spring/Summer '90**. You'll find 3-D rendering, modelling, and animation packages; ray-tracing, painting, and basic animation programs; CAD systems, object editors, clip art, fonts, utilities, tutorials—**EVERYTHING YOU NEED** to make your graphics look just like the pros'!

Pick up your copy of **AC's Guide – Spring/Summer '90** at your local Amazing Dealer!

pin out of VCP200) will light when it recognizes that command.

If you encounter problems try holding the microphone closer to your mouth and speak directly into it. Try all the commands, to check if your encountering a problem with just one or two commands.

Also if you're in a high noise area, that could prevent the chip from responding. In general the quieter the area the better. If all fails recheck your wiring.

GOING FURTHER

The literature with the chip describes a simple latching circuit that can be added to this circuit. The latching will enable you to hold the Go and Reverse with either the Left Turn or Turn Right. This would be the same as holding your joystick in an up diagonal or down diagonal position.

If you do decide to add this I'd advise you to watch your current draw from the joystick port, keep it under the 100 ma. max.

As I stated before, the chip can easily be fooled with non-command words. With a little forethought and testing you should be able to derive your own vocabulary for

the chip. Start with homonyms, words that sound like the command word. Such as release, rehearse, remorse for the command word reverse.

You can increase the effective range of your microphone by increasing the resistance of the resistors R4 and R7 to 470k. This may increase the amount of static you pick up and that could render the circuit inoperative, but you may want to give it a try. You also might invest in one of those headphone microphone combinations, that would keep the microphone a few inches away from your mouth.

OTHER CIRCUITS

You could use this circuit as the front end to some other interesting projects. I wouldn't advise putting this on a model electric car as a substitute for Radio-Control. I could easily envision someone running after the car yelling Stop!! Stop!! The effective range of the microphone is pretty limited.

But a voice-controlled wheelchair for the handicapped is an excellent application worth pursuing.

Parts List

see schematic (Xtal)	10 pf
c2	47 pf (max)
c5	270 pf
c3, c4, c12	.01 uf
c6, c7, c11	.1 uf
c1	.22uf
r2, r5	1k
r1, r15	2.2k
r10, r11	5.6k
r3, r6, r13, r14, r16	10k
r9	15k
r12	100k
r4, r7	330k
r8	10 meg
LM324 Quad op-amp	RS# 276-1711
VCP200	RS# 276-1308
Electret Microphone	RS# 270-092
LEDs	RS# 276-026
XTAL 10 MHZ	*
9 pin male	RS# 276-1427
9 pin female	RS# 276-1428

* 10 MHz crystal can be special ordered from local Radio Shack store. Or from Mouser Electronics
1-800-346-6873
PN# ME332-1100

•AC•

(3-D Pro, continued from page 49)

decisions about where to save files or requiring manual switching of directories via the file requester for each different type of file used.

Speaking of animation, Progressive has included Animation Station, a powerful and versatile ANIM file editor, also available separately. Animation Station is a storyboard-based editor, giving you an optional comic-strip-like display of all frames in miniature. You can move, copy, insert, and delete frames and ranges, reverse ranges of frames, and effect frame-by-frame palette and timing changes. You can save all or selected frames as IFF images, or convert a sequence of IFF pictures into an animation. Special effects include halving or quartering the size of all or selected frames with placement anywhere on the screen, flipping, motion blur, and scrolling. And that's not all by any means. Once you start using Animation Station you'll find uses for it you might never have dreamed of—it's that great.

All in all, the 3-D Professional package is most impressive. Despite the fairly high cost, I recommend it highly to all Amiga users interested in three-dimensional graphics and animation. Its ease of use and extensive tutorial material make it especially ideal for beginners. The extraordinary amount of detail that went into its design marks 3-D Professional as a program with a long life both behind and ahead of it—indeed, it was three years in the making.

•AC•

3-D Professional
Price: \$499.95
Progressive Peripherals & Software
464 Kalamath Street
Denver, CO 80204
(303) 893-6938
Inquiry #221

HOW PRO IS 3-D PROFESSIONAL?

by Frank McMahon

AT WCTV, THE CABLE TELEVISION station in Rhode Island where I work as Production Supervisor, we've used a number of products produced by Progressive Peripherals & Software. From the FrameGrabber digitizer to the ProGen genlock to CLIMate, Progressive has consistently put out quality products. So I must admit I was more than a little anxious pending the release of 3-D Professional. We've always used Sculpt-Animate 4-D for all our 3-D rendering. Sculpt has advantages such as ease of use and quality output, but falls short in lack of textures and poor lighting control. Also the fact that Byte by Byte no longer supports the series and has instead decided to concentrate on the Mac hasn't helped either. Good software needs to be constantly updated and revised and Sculpt-Animate 4-D has been in a holding pattern for about a year now. While I've used Turbo Silver on occasions, its complex interface and features demand more of a learning curve than we currently have time to devote. (This hopefully will be cured with Impulse's Imagine.) However, there is no denying the knock-out ray traces that Turbo Silver is capable of producing.

So is 3-D Professional the answer to our TV needs? Well the answer to that requires knowledge of what exactly our demands are on a daily basis. We do need a quick and easy way to produce 3-D graphics and animations on a weekly

schedule. 3-D Professional's interface is built for speed. It's easy to get around in, and quite logical in orientation.

One of the things we do most in 3-D is create logos. We output them to our Mimetics FrameBuffer in 24-bit mode and record them to 3/4 tape. To test out the program I tried several different logos that had been predesigned using Sculpt 4-D. 3-D Professional will load many formats of 3-D objects, everything from Videoscape 3D (GEO) to Forms in Flight to Turbo Silver to 3Demon. This makes it easy to not only use preexisting objects and scenes you currently have, but to take advantage of the many object and data disks currently on the Amiga market. The objects I imported from Sculpt 4-D also brought in their own 24-bit color information which is passed on to 3-D Professional. 3-D Professional handles color as "direct" and "indirect". The "direct" color of each object (from a palette of 16 million) is transferred, and the "indirect" color (from the Amiga's standard 4096) is matched to the closest color in the standard palette, so I can render in true color in either 24-bit or any of the standard Amiga bitplanes.

Also, in most of the different formats the properties of the object (metal, shiny, dull) are also transferred in to the program. Sculpt textures work out OK (although Glass2 "true glass" does not seem to be supported), however Turbo Silver users will find,

due to the infinite amount of textures possible, that their surface settings will be stripped once the Turbo object is loaded in. However, most of the textures can be easily set once the object is loaded in using standard numerical sliders to which Turbo users are accustomed.

It should also be noted that 3-D Professional does not save in either Turbo Silver or Sculpt format. The reason is, once an object begins loading, 3-D Professional doubles the object's polygons to conform to its object standards and to avoid potentially hidden surface problems. I found out after I began importing logos that this can make for some really complex objects. This procedure is really only needed if you are using "smoothing" however, so if an object is made of hard edges it is not really a concern. The logos can be loaded in 2-D and extruded from within the program, which helps keep down the polygon count. The program currently does not have true anti-aliasing, which is hopefully in the works for the next update. Because we render to a 24-bit board, which uses the highest Amiga resolution (746 x 484), the smoothing calculation is not usually needed. However anti-aliasing in 24-bit (as in Sculpt 4-D) works magnificently when rendering because it has so many colors to work with. Anti-aliasing usually is needed more in HAM mode.

Creating logos has proved to be quite easy in 3-D Professional. A text creation tool is provided which allows you to select any Amiga-compatible font and just type in what you want to say. Justification and point size, as well as style, can all be set. Some logos we produce are variations on existing fonts, so being able to load in the original and alter it in 3-D is a big plus. 3-D Professional also converts any brush to a 3-D object, so 3-D Professional even allows for the import of Deluxe Paint III objects! The only drawback is that each pixel is converted to a four-vertex polygon.

This seems a bit too much calculating sometimes so it's best to keep the brushes small. At the station, we always need to render and produce as quickly as possible, so we run 3-D Professional on our 2500/30 with 5 meg.

Once I create a logo, the first thing I usually do is test out the different textures. This is where Sculpt 4-D gets left in the dust. 3-D Professional comes with

several built-in textures which allow hundreds of different combinations by changing the colors, shape of the texture, thickness of texture or lines, transparency, roughness, refraction, reflection, specular reflection, glossiness, and more.

Using our Mimetics FrameBuffer board we can take advantage of the ability to choose any of 16 million colors, since 3-D Professional's texture window has color sliders that allow the user to set 255 shades of red, green, and blue rather than the standard Amiga sliders which jump 7% from one color to the next.

In the past week I've rendered several different textured objects and have had excellent results. Favorites include marble, brick, and terrain. Terrain is actually meant to shade (using 4 base colors) from top to bottom a landscape (which 3-D Professional also generates effortlessly): white snow caps, brown mountains, green valleys, and blue lakes. However all colors are adjustable and I've been using Terrain for anything but land scenes.

By using combinations of green and red and purple, for example, I've been able to come up with dazzling multi-colored objects. Chrome sheens and glints of light across a roll of columns are rendered perfectly and look startling in 24-bit mode. This sort of multi-colored rendering is used constantly at high end 3-D houses. One thing about the many different textures is that they are true textures, which may cause problems during animations. Instead of the program wrapping a concrete texture on a brick, for example, it actually "carves" a brick out of a concrete universe, like a cookie cutter. The result is amazingly realistic and there is no problem panning a camera around it during an animation. However if your camera is fixed and your object is set in motion, your object will "swim" through the textured universe and the surface of the object will constantly change. This may make for some excellent special effects but is not practical for most situations. Marble is another texture I have used quite a bit. For my money, there is nothing more professional than a 16-million-color image cast in STONE. The classy appearance of marble cannot be beat; even if the logo or title I'm producing is not made of marble, I always try to

include at least a marble platform or a row of marbled balls. Different colors can be used which can simulate any kind of stone. The other day I created a killer emerald tablet. Bricks are fun too, and being able to adjust the roughness of the surface makes for some realistic-looking brick walls.

After using 3-D Professional in the studio, I'm convinced it's going to be a big part of our graphics future. We've always hoped a 3-D program would come along that would produce outstanding results with an easy-to-use interface. 3-D Professional is it. It reminds me of Deluxe Paint III a LOT. Easy and fun to use; and the more you experiment and try combinations of different commands, the more impressive results you can achieve.

Whether you are planning to purchase 3-D Pro for use at a television station, or just for home use, there is one piece of hardware that is a must-have: the Mimetics FrameBuffer.

Since we purchased the FrameBuffer late last year, we've used it constantly with Sculpt 4-D, and now with 3-D Professional. 3-D Professional fully supports the FrameBuffer. It's even on the output menu.

The Mimetics board includes a video-in and video-out and allows your Amiga to display renderings of up to 16-million colors in hi-res interlace overscan. Words cannot describe the difference that this board makes to producing quality 3-D renderings. The output is stunning, to say the least. What's best is that the Mimetics FrameBuffer board is very low in price (approximately \$500-\$600), and is easily installed. For the Amiga to compete in the television graphics world, 32 colors, or even 4096 colors, just isn't going to cut it. If you are serious about 3-D graphics, this board is a must.

And 3-D Professional is a must. At our studio I've used it to produce some excellent graphics and animations quickly and easily. This is not the kind of program that bogs you down in numbers, options, and commands. Its raw power is pumping below its well thought-out and designed user interface. 3-D Pro makes creating 3-D output a breeze. Given the hectic nature of television, it's exactly what we need.

•AC•

T H E G A M E O F • H A R M O N Y •

by Joe DiCara

WHAT'S IT ALL ABOUT

By definition, harmony is the peaceful coexistence between two or more objects or entities. That's simple enough, but the actions necessary to obtain a state of harmony are at times complex and contradictory. So too is the game of Harmony. This game is in the same category as Shanghai, Tetris, and Kikugi; a strategy puzzle. At first glance these games seem extremely simple, even childish. Yet upon closer examination and involvement they become complex and very challenging, even diabolical. How can this be? Because they devour large quantities of your time. You know what I mean. A game is started just to pass some time while you wait for a download to finish or that spreadsheet to print

out. What's the result? The "just one more game" syndrome, that's what!

A POD IS BORN

Accolade advertises Harmony as a game of challenge, that rewards a relaxed approach and a calm state of mind. The objective of the game is simple. You maneuver a spinning sphere around and among other colored spheres, nudging together any of similar color, thus establishing a state of Harmony. But, if you accidentally allow or cause two different colored spheres to come into contact, well as they say, your problems are just beginning. It seems when these two spheres touch, a third uniquely colored and smaller sphere, called a pod, is created. Now two things

can happen. First, if you manage to glide over this new pod your energy level is increased. But, if you ignore or can't get to the pod soon enough (there might be other spheres, objects, or more pressing tasks at hand) it grows into a full sized sphere. To add to the pressure, as time passes the spheres begin to pulsate. If enough time passes they go completely bonkers and explode, which costs you energy. Simple huh? Relaxing even! Right.

Now to make things even more interesting some spheres are attached by an elastic cord. At first that seems like an advantage, but if you bump a sphere with too much force in the wrong direction, the cord jerks the attached spheres in directions you had not planned. The result -

• F U T U R E W A R S •

C a n Y o u S a v e H u m a n i t y ?

by Miguel Mulet

Life can be complicated, no matter what your profession. Imagine a poor window washer, who one day tires of his task of cleaning the windows on a large skyscraper. Thus, the poor chap wanders into his boss's office, and gets more than he bargains for - a trip into the past, as well as into the future!

"Future Wars - Adventures in Time" is a new adventure game introduced in the United States by Interplay. Apparently, the game has won several awards in France (its country of origin), and after playing the game, it is probably deserving of these awards.

In "Future Wars", you assume the role of the window washer, who really has no idea what the future holds for him. (If you don't want to spoil the plot, skip this

paragraph and go on to the next one!) Our hero stumbles onto a time machine, which transports him into medieval times. While there, he learns of a race of aliens who plan to take over the Earth. The only way he can stop them, however, is to travel through time, destroying their plan before it is put into action.

Although the plot may sound far-fetched, it remains fun and interesting throughout game play. All the graphics are well done, although some scenes occupy less than 1/4 of the screen. At first this seems strange, but it gives the player an idea of the scale of the world surrounding him. The excellent graphics are accompanied by digitized sounds of everything from footsteps to machinery, so you may want to turn the volume up on your stereo. Speak-

ing of stereo, there are several parts of the game featuring a stereo soundtrack, which greatly enhances gameplay. (The box claims that the complete musical score is available on compact disc, although there are no instructions on how to order it).

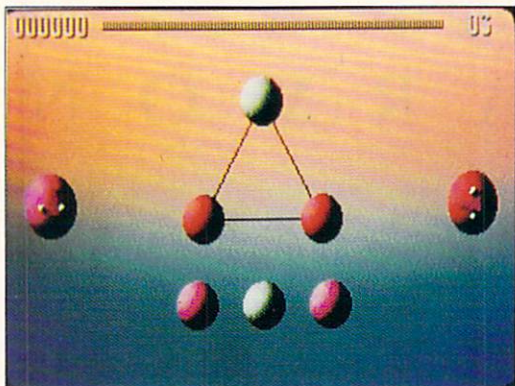
Maneuvering through "Future Wars" is performed solely through the mouse - there is absolutely no typing involved. To move your character, all you need to do is move the cursor to the location to which you would like to go and push the left mouse button. The right mouse button serves to bring up your "ACTIVATION MENU". This menu allows you to examine objects, take objects, use objects, and speak to other characters. You can also get an inventory of the objects you are carrying. The "OPERATE" command in this menu allows you to

(Harmony continued)

more pods. There can be as many as three or four spheres joined by a cord, and at times even your own sphere might become attached. To further increase your gaming pleasure certain levels have barriers of different shapes and arrangements placed randomly on the playing field for you to maneuver around. Again this all seems to be no big deal, but just try it - diabolical I tell you!

There are 50 screens presenting different combinations and variations of spheres and barriers to challenge and hopefully sooth your nerves. As you probably can imagine, most of the screens will take skill and much practice to clear successfully within the allotted time. Fortunately the timer that causes the spheres to pulsate can be shut off allowing the player all the time needed to plan moves and strategies. This mode, called Mantra, is relaxing, fun, and challenging. Anyone can eventually bring about the state of Harmony, but how many movements will it take?

Concerning game play there is very little to criticize or improve upon. The control of your sphere and its action on the wrap-around playfield does take some



practice, but can be quickly mastered. While graphics are in Amigas' low resolution mode the program uses the complete 32 color palette to produce pleasing backgrounds and nicely textured spheres and objects. Music-like sound effects are well done and enhance game play.

Harmony is not copy protected but herein lies the only complaint. Accolade has begun to use dark red paper code sheets (red because it can not be easily photo-copied). The program asks you to

match an opening game screen with one printed on the code sheet. For most of their games the shapes or patterns of objects can be readily discerned. Unfortunately Harmony screens do not lend themselves well to this process. There are no colors to distinguish one screen and its objects from the next. It is very difficult to select the correct screen. This is not fatal, its just very annoying.

Harmony is a pleasant departure from the standard blast'em and smash'em stuff. So the next time you have a few minutes to kill fire up Harmony. Just don't get to upset with yourself when you can't shut it off. Diabolical, simply diabolical.

•AC•

The Game of Harmony
Accolade
550 S. Winchester Blvd.
San Jose, CA 95128
(408) 727-0256
Price \$44.95
Inquiry #218

(Future Wars continued)

operate almost anything operable on the screen. Instead of typing "Open door" you would select "OPERATE" and then move the cursor to the door and click with the left mouse button.

There is only one other menu, and it appears if you press both mouse buttons at one. This menu is the "USER MENU". It allows you to restart the game, select the backup drive, and load or save a game. The game is paused any time you bring up either menu.

"Future Wars" is provided on two disks, which may be copied (the manual encourages you to play from backup disks only). It will run on an Amiga 500, 1000, 2000, or 2500 with 512K and Kickstart 1.2 or above. A blank disk or hard drive is required. Copy protection is in the form of a color picture on the back of the 5 page manual. After the game is booted, a screen which is a black and white duplicate of the back of the manual appears, asking



you to identify the color of the highlighted area. You are asked to do this twice, after which the game proceeds to boot. If you're color blind, you may have problems with this scheme.

"Future Wars" was moderately challenging, as well as fun to play. The player must look carefully at each scene, as some objects are only 1-2 pixels in size. Other objects only appear momentarily, and if

you're not looking at the screen at that time, you'll miss an important item. Although the plot was interesting and had several interesting puzzles, it only took about twelve hours to get through the entire game (and I consider myself an "average" adventurer).

Even though the game was short, I thoroughly enjoyed it and felt it was worth the price. "Future Wars" features good graphics, sound, and game play, combined with an entertaining plot. Interplay states that this is the first in a series of this type of adventure game, and hopefully its descendants will be even better.

•AC•

Future Wars - Adventures in Time
Interplay Productions
1575 Corporate Drive
Costa Mesa, CA 92626
(714) 549-2411
Price: \$49.95
Inquiry # 219

PRESENTING

WORLD OF AMIGA IN CHICAGO

★ *Starring* ★
THE AMAZING AMIGA

★ *Featuring* ★
**Amiga Hardware • Amiga Software
Amiga Accessories • Seminars • Bargains**

**Rosemont O'Hare Expo Center,
Rosemont, Illinois**

October 5-7, 1990

Friday, Saturday & Sunday 10am-5pm

Pre-registration:

\$8 per day or \$20 for 3 days

Deadline for pre-registration Sept. 18

Registration at show:

\$10 for 1 day

\$25 for 3 days

Registration includes exhibits and seminars.



**WORLD OF
AMIGA**
IN CHICAGO

FOR MORE
INFORMATION
Call (416) 595-5906
Fax (416) 595-5093

SAVE WITH PRE-REGISTRATION

Please register me for the 1990 World of Amiga in Chicago at the Special Pre-registration Rate.

NAME _____

COMPANY (if applicable) _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

☐ One-day registration (\$8) ☐ Three-day registration Oct.5-7 (\$20)

Make Check or Money Order payable to
The Hunter Group, 3380 Sheridan Drive, Suite 120, Amherst, NY 14226

Other upcoming events produced by The Hunter Group include **COMMODORE AMIGA USERS FAIR** in Valley Forge, PA, September 15 and 16, 1990, and **WORLD OF COMMODORE AMIGA** in Toronto, November 30 to December 2, 1990.

Programming in C on a Floppy System

by Paul Miller

IF YOU HAVE AN AMIGA SYSTEM WITH ONE OR TWO FLOPPY DRIVES AND ONE megabyte of RAM or more, and have considered programming in C, you may have been turned off by compiler requirements of two megabytes of RAM and/or a hard drive. This is increasingly the case with today's newer (and larger) C compilers, such as Lattice's version 5.04 system. The purpose of this article is to describe how the one- or two-drive Amiga user with at least one megabyte of RAM can begin developing programs in C. Yes— even with a stock Amiga 500 with the 512K RAM expander.

First, decide what compiler, text editor, and utilities you want to use in your development environment, and how you want to interact with it. For the purposes of this article, I am assuming an Amiga 1000 with 1.5 megabytes of RAM, Kickstart 1.3, ARP 1.3, two drives, and Lattice C version 5.04. (This is the system I currently use.) Modifications for smaller systems will be detailed later.

A FOUR DISK SYSTEM?

Yes, the development system will be a four-disk setup. Don't be alarmed, though! It's really not that bad. The first disk will be your boot disk, and will contain the compiler, linker, libraries, and some support DOS commands to get your environment up and running. I strongly suggest you use the ARP 1.3 commands, as they are much smaller and provide greater flexibility in file manipulation. I have my Preferences set to an interlaced screen (colors set to medium-green text on a black background, with blue and orange highlight colors) too, so I can have two full files displayed at once in my editor. If you don't like the flicker, a non-interlace display should be OK (and require about 32K less RAM; 1-megabyte users might consider this).

The second disk (DISK 2) will hold all of the compiler header (include) files and your most-common DOS commands, such as your editor (I use DME v1.31, Fred Fish 168-169) and any background programs you might want to use.

The third disk (DISK 3) will hold all of your source code, object modules, and compiled programs.

It's true—where there is
a RAM disk, there is a way.

The fourth disk is a RAM disk. (I use the ASDG-rrd RAM disk, FF 241, because it adjusts its size according to how much is in it, rather than RAD: which is always a constant size, regardless of what's in it.) This is initialized with the first disk, and the contents of DISK 2 are placed here during the installation process. Furthermore, all of your source code editing and compiling will take place here. You'll have to get into the habit of copying your modified files back onto your source code disk before you power down!

SETTING IT UP—

DISK 1: THE BOOT DISK

- 1) Make a copy of an ARP 1.3 Workbench disk (a disk with all of the ARP commands and library on it), and erase all of the commands, libraries, devices, fonts, etc. you'll never use. Label this disk "Development" or the like.
- 2) Make an "LC" directory on the disk and put your Lattice LC1, LC2, and BLINK in it. Make a "LIB" directory and put c.o, lc.lib, amiga.lib, and lclm.lib in it (plus any other modules or libraries that you might use and can still squeeze on there).
- 3) Type the startup-files (Listings 1 and 2) into the "s" directory, omitting or altering things to your liking. If you can't get a hold of the ReadKWIK and RiteKWIK programs, you'll have to use a "copy DISK2:#? to vd0: all" command in place of the reference to ReadKWIK. (This shareware program, written by Gary Kemper of Tigress, bulk copies an entire disk of files in a special format, and quickly.) In this case, DISK2: would be the volume name of the disk containing the headers and commands (see below).
- 4) Put the asdg.vdisk.device into your devs: directory and modify the Mountlist accordingly (I have my HighCyl value set to 107, about 880K of RAM disk space; if you have under 1.5 megabytes of RAM, you'll want to use a smaller value here). If you prefer a different RAM disk, make sure to change references to this one accordingly.

DISK 2: THE COMMANDS/HEADERS DISK

- 1) Mount the RAM disk and create three directories in it: "c",

"t", and "include". Copy your most commonly used DOS commands into the "c" directory, including your text editor, background programs, "LC" from the Lattice disks, and any scripts you would prefer to run from RAM.

- 2) Copy the compiler header files from the Lattice disk into the include directory. If you have only 1 megabyte of RAM, use the compressed headers only.
- 3) Use RiteKWIK (if you can find it) to copy the contents of your RAM disk onto a blank disk (or use "copy vd0: to DISK2: all" after formatting a disk and naming it "DISK2").

DISK 3: THE SOURCE CODE DISK

Format a disk, and set it up to conveniently store your source code and object modules on it. For each large project I have a separate disk with directories labeled "Source", "Object", and "Scripts". In the scripts directory I have a file called "Environment" that assigns all of my directories and sets up the environment for that particular project. For example, my "Environment" file assigns the device WORK: to the "Source" directory, OB: to the "Object" directory, PIC: to a "Pictures" directory (if a particular project requires them), etc. At the end of my startup sequence is an Execute command that calls this script and automatically sets up the environment for any project I want to work on.

INSTALLING THE SYSTEM

To start up your development system run, boot up with DISK 1, and have DISK 2 in the second drive. If you only have one drive, and are using ReadKWIK, wait for it to pop up an "Incorrect Disk" requester, then replace DISK 1 with DISK 2 and click "OK" (don't forget to change the reference from drive 1 to drive 0 in the startup script). If you're not using ReadKWIK, wait for an "Insert volume DISK2:" requester to appear. When the files from DISK 2 have been copied to RAM, remove DISK 2 and replace it with DISK 1 (if on a one-floppy system, so the rest of the startup script can continue) or DISK 3, in the external drive. Now you can get to work.

BRIDGEBOARD USERS!

Don't waste money, slots, or desk space buying extra IBM-compatible or Amiga floppy drives! The **Bridge Drive Commander +** gives you direct access to all your internal and external Amiga drives from the Bridgeboard, and direct access to IBM type 360K and 720K drives from AmigaDOS. **Bridge Drive Commander +** is totally transparent and automatic. Put an IBM type disk in any drive and use it just like on any IBM compatible! Put in an Amiga disk and return to Amiga use! Just that simple, just that fast! One drive can use Amiga disks at the same time another is using IBM-compatible disks. Disks are completely usable by other Amiga and IBM-compatible computers. All hardware, no software drivers to load, no precious memory or expansion slots used up. Plugs onto motherboard at internal drive connector. (No soldering or wiring changes.) Compatible with all Bridgeboards (8088, 80286), SideCar, all accelerator boards (any 680x0), hard disks and other hardware and software.

Bridge Drive Commander + \$ 97.50

MJ SYSTEMS

Dept 10A, 1222 Brookwood Road, Madison, WI 53711

1-800-448-4564

(24 hours MasterCard/VISA)

Product names are trademarks of their respective companies.

Circle 149 on Reader Service card.

FUNCTION KEY MACROS

If you can get a hold of a function key macro program, you can easily customize and simplify your compiling process. For example, I have function key F1 bound to the macro "run Dme", so all I have to do is hit F1 and type the filename to edit a file. The same goes for F3, which expands to "execute c/compile". After I type the filename to compile, this executes the "compile" script in my "c" directory, which calls LC with my favorite parameters. Pressing F4 enters "Blink with MakeFile^m (^m is a carriage return)", which invokes the linker (I have a MakeFile in RAM already configured for each particular project).

There you have it. I've tried to make this as flexible as possible, with a minimal number of disk changes. Two-drive users should only have to do one swap (switching DISK 2 with the source code disk in the external drive, after DISK 2 is copied to RAM). One-drive users will have to swap the boot disk with the source disk between compiles, but since most of my time is spent editing anyway...

You might have to custom tailor the number of background programs and the size of your RAM disk to fit comfortably within your memory restrictions (remember the compiler uses over 100K of working memory per pass). On my 1.5-megabyte system with five background programs and uncompressed headers, I have about 400K of memory leftover to do everything else in. A one-megabyte A500 with compressed headers and one or two background programs should have about 400K as well. Not bad for a "minimal" configuration. It's true—where there is a RAM disk, there is a way.

Listing 1: startup-sequence

```
c:SetPatch >NIL: ; 1.3 Kickstart
Patch program ; use FAST RAM
c:FastMemFirst ; your
before CHIP RAM ; 1.3
c:RTClock ; 1.3
clock program goes here ; 1.3
c:FF > NIL: ; 1.3
FastFonts utility ; make the Resident and ARP
Resident c:Resident pure add ; resident
Shell ; mount the ramdisk
Resident L:ASH SYSTEM pure add ; initialize
and console ; initialize
Dir vd0: ; initialize
ramdisk ; initialize
NewShell Window NewCON:0/0/640/400/AmigaShell From s:install-
system ; open a new Shell
EndCLI >NIL: ; open a new Shell
and end this one
```

Listing 2: install-system

```
IF EXISTS vd0:c/DME ; Check to
see if the ramdisk ; is empty
SKIP FROOD ; if a
by checking to see ; if a
EndIF ; if a
file is still there. ; If it's empty,
Echo "More than one moment, please..." ; If it's empty,
copy Disk 2 ; (head-
Echo "" ; fast
ers/commands) into it. ; fast
ReadKWIK 1 vd0: ; fast
disk file copier (FF??) ; fast
LAB FROOD ; fast
Path ADD vd0:c df0:lc ; add
various command paths ; change
CD vd0: ; change
to ramdisk directory ; Run your
Run c/GOMF ; Run your
various background ; programs
Wait 1 ; programs
and utilities. ; Function key
Run c/FuncKey s:func.key ; Function key
macros, FF106 ; Function key
Wait 2 ; Screen/
Run c/ScreenX ; Screen/
clock utility, FF158 ; Screen/
Wait 1 ; Screen/
Run c/Qmouse >NIL: -Fs:Qmouse.cfg ; Mouse utility, FF49
Wait 2 ; Mouse utility, FF49
Run c/Blitz -b ; Back-
ground file viewer, FF60 ; Back-
Echo "Assigning Device Names..." ; Back-
Assign A: DF0: E: VD0: ; put your
devices here ; put your
Assign LC: df0:lc/ LIB: df0:lib/ INCLUDE: E:include/ QUAD: E:
Assign t: E:t/ ; put your
Assign DOC: df0:Docs/ LOG: df0:s/ ; put your
alias clear echo "E[0;0H*E[J" ; put your
alias reverse echo "E[0;0H*E[41;30m*E[J" ; put your
alias normal echo "E[0;0H*E[40;31m*E[J" ; put your
Stack 10240 ; put your
Echo "(Whew!)" ; put your
Echo "" ; put your
Echo " AmigaShell 1.3: Lattice AMIGA 'C' 5.04 Development
Environment" ; put your
Echo "" ; put your
Dir ; put your
Execute df1:Scripts/Environment ; execute anything else you
want now
```

NOTE: You can use PowerPacker (FF 253) to decrease the file sizes of your background programs and DOS commands to conserve disk space, if you can sacrifice disk-loading time.

•AC•

upgrades

fixes

updates

new
releases

bug bytes

by John Steiner

LETTERS FROM READERS HAVE dominated the column over the last few issues. I always enjoy reading the mail, however, I am having to become selective due to space limitations about how much material I can put in each column. If you have written, and you don't see your letter mentioned here, there are a few reasons why I might not have used it. First of all, remember there is a two to three month publishing delay between the time I get a letter, and the time you see it in print. If you send the mail electronically via CompuServe or People Link, the delay is cut to the minimum.

Secondly, I try not to pick on manufacturers regarding software bugs. Therefore, if you point out a bug that you've discovered, and have no workaround for it, you will only find it mentioned if there is a danger of loss of data or work on the part of the end user. Remember in many cases, one person's "bug" is another person's "feature." I tend to make an exception to my rule about not having a workaround if the problem appears to be major and the other readers might have stumbled across a workaround for it. Your letter is also more likely to get mentioned if the reader tells me that he or she has brought the problem to the attention of the manufacturer's technical support people, and they refuse to respond or make little attempt to reply to the reader. Remember, in many cases it is difficult or

impossible for a software developer to fix some problems in a very timely fashion. Computer software is complex and with the models of Amiga systems proliferating, and the impending changes to the operating system, program bugs are often hard to find, and hard to remove.

I often get letters from people regarding the way a program operates. Though the implementation of a feature in a specific fashion may well be an irritating problem to one individual, it really isn't a "bug". Clearly the software was written that way, and changing the way the program operates may or may not be of benefit to the other users. In fact, this month I also received correspondence from Stanley Skirvin of Scottsdale, AZ on just this topic. He commented on the negative effects of changing the way a program interface appears to the end user when making minor upgrades. Undoubtedly, Mr. Skirvin would likely be unhappy if the developer were to change the way something is being handled at the request of only a few users. He went on to suggest that user visible changes should be introduced only if they represent identifiable improvements or correct errors perceptible to [a large number of] users.

I have to agree with Mr. Skirvin. In fact, I'd like to see more standardization of the user interface in general as it makes the learning curve easier to manage. I know that I will probably hear from those who

feel that standardization stifles creativity, so I will add that innovation with regard to the user interface is always welcome if it improves the performance and productivity of the application. There are indeed many examples of that kind of innovation improving the general "look and feel" of a program. In any case, I will avoid reprinting comments on how a particular function in a given program operates unless you have found a much simpler, or more powerful "workaround" to complete the designated task much more productively.

I received a letter via CompuServe electronic mail from Anthon Pang with several comments regarding Gold Disk's Transcript word processor. He bought Transcript version 1.0 for heavy duty document production.

He warns that users should be cautious of rapidly switching from edit mode to screen preview mode (where they open a scrollable custom screen) as it appears to fragment memory. Eventually, the preview mode will not function, or the system will Guru.

Mr. Pang comments that the "hr" code, to set a header on right-side pages, does not work. He provides the following workaround. Instead use a combination of "hb" (header both-sides), set to the intended header text, and "hl" (header left-side), to replace/cancel the left-side effects of the "hb" code. The same fix applies for

the "fr" (footer for right-side pages). Use "fb" & "fl".

Transcript doesn't handle more than 1 space between words or sentences (for example, after a period) properly, in the case where the next word or sentence wraps to the next line. The additional space(s) will cause the text to be indented. His workaround is to simply adjust the window size to match the final output text width, to locate the unwanted occurrences. Another workaround I would suggest is to use Global Search and Replace to replace all occurrences of the double spaces with single spaces. Be careful of this option if you have columnar material or indent paragraphs with tabs, as tabs are represented as spaces in Transcript.

Mr. Pang also provided several tips for using Transcript effectively: 1)When using the "ep" (eject page) code to end a page and start a new one, if it appears at the end of a page, a blank page will be generated. 2)Pages must be in sequential order, as you cannot use the "pn" (set page number) code once set within a document, to "jump around". 3)And with virtually any program, use the Save command frequently when working with important material.

Another electronic mail reply regarding M2Sprint was received from Mr. Tom Gist, the person who originally reported a problem with the longreal function in the program. He restated his comment that he had problems with the bug under both the original 1.0 and with version 1.12. He did comment that he would double check his installation of 1.12 to make sure he got it completely installed correctly. Other readers have commented that there is no problem with longreals in 1.12, but they had problems in 1.0. If you are having problems with version 1.12, drop me a note and I will let Mr. Gist know that others are having the same problem.

I don't know if this column had anything to do with it, but Intuitive Technologies, having taken a lot of heat in these pages for poor technical support, has hired a full time technical support person. Bruce Brandt is the new technical support engineer at Intuitive Technologies. He said in a posting on People Link that he will be available in the office from 10 AM until 4 PM

every day, Monday through Friday. The phone number he left in his posting was 408-646-9147.

Everyone complains when service is bad, and little is said when service is good. Let me know if their support services improve, I'll pass that information along as well.

I received a letter from a D. Reimer of Aurora, CO who comments that he just bought the Toolbus Expansion Box from Applied Technologies and an 8-UP! board from MicroBotics with 2 meg of RAM for his Amiga 500.

He is having a problem when he powers up his A500 with the Progen genlock hooked up, the power comes on but the computer doesn't boot. The only way he can get it to boot is to disconnect the Progen, power up the CPU, disconnect the monitor cable, plug in the genlock and then plug the monitor into the Progen. He is wondering if a different genlock will solve this problem or if it is built into the A500.

Mr. Reimer fails to mention whether or not the Progen works properly if the extra memory and expansion toolbox are not connected. I am not very familiar with either of these expansion accessories, however I do know that the Progen gets its power from the A500 power supply. If the other accessories are also getting power from the A500 supply, it could be that the power supply is being overtaxed. Also, it is highly NOT recommended to connect and disconnect peripherals from Amiga ports with the power connected. I can say that I've done this myself in the past, and I can also say I have lived to regret it. In any case, those are the only suggestions I have regarding this problem. Anybody else have any suggestions as to what Mr. Reimer's problem is, and how it might be solved?

Another hardware problem was reported to me in EMail by Rawli Puig. He has an Amiga 2000 with 3 meg of RAM (two meg are installed on a Spirit 8-UP! board), IVS Trumpcard 48 megabyte Seagate hard drive, IVS Printerface and 1 meg Agnus. The unit was assembled at the store where it was bought. Everything works great with the exception of Digi-View 4.0 (gold). The program will not save files at all. Attempts to save digitized pictures cause the pointer

Icon to turn to wait state and remain there. He has called IVS, NewTek and Commodore with no success. He comments that this is practically the same system he had before in his Amiga 500. He was using a A590 Commodore Hard Drive instead of the Trumpcard. He has also noticed that some programs open different windows on the 2000 than the 500. These are the same programs from his A590 which were backed up with Quarterback and reinstalled on the 2000.

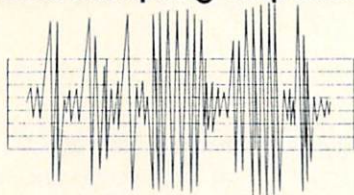
In a follow up letter, Mr. Puig commented that the problem appears to be related to the parallel port on the motherboard which seems to have a bus problem, along with the external disk drive port. After changing the CIA chips (done by a Commodore Service tech) and running Digi-View again, nothing changed. Another symptom of the problem occurred when an external disk drive was attached, every disk that was placed in the drive was labeled BAD. He commented that the service tech mentioned to him that there seemed to be a lot of problems with the 2000 boards. A technician in our local Amiga dealership here in Fargo ran across the bad drive problem on one of his customer's new A2000HD. Any drive connected to the external drive port failed to function with the same symptoms that were listed here, any disk placed in df2: were labeled BAD. He traced the problem to a defective solder connection on the motherboard in the external drive port connection area.

Whether or not this problem is related to Mr. Puig's is up for speculation right now, but it appears there might be some quality control problems related to the 2000 motherboard.

Roger LeVine sent EMail regarding Professional Data Retrieve. He wrote to report a bug in the printer mask editor and a workaround for it. "The manual states that if you place field names in the printer mask editor you will get a printout columnar in type, that is, each field will print to its maximum size as set in the screen mask editor, regardless of the length of the data in that field. To print only the data entered, that is "John Jones" rather than "John Jones ", you should enter the field names as

AudioLink

**16-bit Linear Stereo
Audio Processor with
Sound Sampling Capabilities**



Beta Unlimited
87 Summit St. Brooklyn, NY
11231

Circle 126 on Reader Service card.

a string, as follows: 1) click on the string gadget 2) type in the field names you wish to appear on the given line within the string

ex. [first_nm]+" "+[last_nm]

3) size the string to the maximum allowable length using the mouse just as you size a field in the screen mask editor.

"The bug is at step 3. No matter what you do, the string "snaps" back to the length of the field names. That is, the above example will print up to, but no more than, 24 characters of the combined first and last name. Anything longer is truncated."

"The work-around for this is to add a string of spaces at the end of the field. That is, if the maximum combined length of first and last names is 40 characters the string you enter should look like:

[first_nm]+" "+[last_nm]+" "

The quotes and the "+" signs count. The trick is that the spaces must be preceded by a "+" sign, and enclosed in quotes. Also, the entire line must be entered into the string before you exit it by hitting return. In addition, the quoted spaces must be the last thing in the string. You can then move the string about the screen to align it as you see fit, but the length is fixed.

"Using this method I was able to print a full set of mailing labels with no missing characters or line overflow."

A posting on People Link pointed out that MicroSystems Software has sent upgrade notices to registered users of excellence! that it has been upgraded to version 2.0. Included with the announcement was a list of 32 "Features and Enhancements".

The program operates faster, user alterations to the Page Setup Requester and the Print Requester can be saved as user-defined defaults, and Headers, Footers, and Footnotes can be cut, copied and pasted, among many other features. A conversion program is provided to enable licensees of Professional Page to use those fonts with excellence!

NewTek, Inc., 215 SE Eighth Street, Topeka, KS 66603 (913) 354-1146. Inquiry #210

Sunrise Industries, Los Gatos, California, (408) 354-3488. Inquiry # 211

According to a posting left on People Link, Version 5.0b of the Manx compiler is out. The upgrade does not contain any significant new features, but rather fixes problems in 5.0a, and is free to all registered users of version 5.0a.

The upgrade is available from the Manx BBS or by calling Manx asking for disk 5.0b update

Manx Software Systems, P.O. Box 55, Shrewsbury, NJ 07702, (201) 542-2121. Inquiry #212

That's all for this month. If you have any workarounds or bugs to report, or if you know of any upgrades to commercial software, you may notify me by writing to:

John Steiner
c/o PiM Publications
P.O. Box 869
Fall River, MA 02722-9970

...or leave EMail to
Publisher on People Link
or
73075,1735 on CompuServe

•AC•

Ham It Up! (v. 1.01)

- ▲NEW! "The Blender" blends and saves color brushes fast!
- ▲Works with DigiPaint™ and DeluxePaint™
- ▲Sixteen charts of 256 colors each
- ▲RGB & CMY values given for each color
- ▲Takes the guesswork out of color selection



Displays and prints all 4096 Amiga colors!

\$39.95* Includes shipping & handling in U.S.
Call or send a check or money order to:
▲Delta Graphics ▲ 48 Dighton St.
Brighton, MA 02135 ▲ (617) 254-1506
*Mass. residents add \$2.00 sales tax
Dealer Inquiries welcome

Circle 118 on Reader Service card.

According to the posting, the cost of the upgrade is \$39.95 plus a minimum of \$4.00 for UPS Ground shipping. If you haven't sent in your registration card yet, you should do so if you want to upgrade.

Micro-Systems Software, 12798 Forest Hill Blvd Ste 202, West Palm Beach, FL 33414, (407) 790-0772. Inquiry #213

Both Great Valley Products and NewTek have relocated. Contact these companies at their new address:

Great Valley Products, 600 Clark Ave, King of Prussia, PA 19406, (215) 337-8770, FAX (215) 337-9922. Inquiry #209

Have your photos, schematics, line drawings, company logos, any 2-D or 3-D material

digitized by

DIGITAL FORMATIONS

use your digitized images in paint or desktop publishing programs.

- All resolutions available
- 2-4096 colors
- 100 % IFF compatible

Rates are \$8.00 for first image and \$4.00 for any additional ones.
(Plus \$2.00 S&H. Originals returned.)

Send orders to:

DIGITAL FORMATIONS
P.O. BOX 3606
Lakewood, CA 90711-3606

Circle 124 on Reader Service card.

Stock-Portfolio

by G. L. Penrose

THIS PROGRAM IS NOT FOR EXPERTS. IT IS, IN FACT, AS unsophisticated a program as you will find anywhere. I designed it to record, organize and track the success of stock and financial transactions; you may wish to use it to catalog your audio- and/or videotape and disk libraries, or perhaps for the maintenance of important telephone or mailing lists.

Although I have personally been involved with programming since the days of the PET 2000, the Amiga and its use of the BASIC language is somewhat different. Believing there to be many like myself who like to type in programs but are *not* experts, I made this program as down-to-earth as possible. There are no fancy algorithms to puzzle over, just simple AmigaBASIC statements to follow.

You are first asked to input a filename. Do this without the .rel extension; the extension is added by the program and can be used in any modification. For instance, suppose you wish to take a list of files. The .rel extension enables you to select only those files, and no others. With a simple loop, you can then list them on the screen, eliminating the bother of having to quit the program to look at the Directory.

Next, the screen upon which all activity takes place is set up (it is never referenced again). The section labelled start: sets up the menu called STOCKS and fills it with the items you will need to use. This is standard procedure, and needs no detailing.

The whole program is based upon the simple use of a relative file. Unlike the 64/128 versions, a single opening of a relative file in AmigaBASIC will allow you to both read to and write from the file. Point of fact, there is really no need to close the file until all activity has ceased. However, in the interests of good programming, the old admonition of closing a file as soon as one operation is complete has been followed here. You will find a Close

1 after every use of the file. The protocol for opening a file is as follows:

```
OPEN "r",#1,NM$,x
```

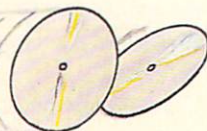
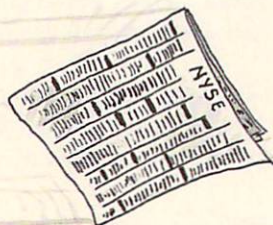
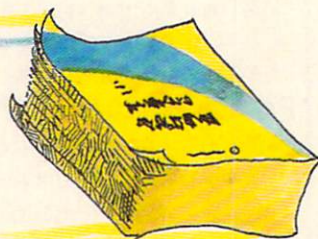
where the lower case "r" (in quotes) represents a relative file (it will not work with an upper case "R"). The #1 indicates the reference number of this file and this file alone. Naturally, it could be any number. The NM\$ is placed there when you input the file you wish to work on. The program itself will attach a .rel to the end of the filename, simply so you can recognize relative files in your directory at a later date.

The x is quite important: it is the total of the field lengths which go to make up each record in the file. In other words, it represents the record length in bytes. This is made up any way you wish. However, it has one proviso: items which are longer than the field length are truncated. As a corollary to this, the field is padded with spaces when an entry is shorter than the field length. The following line shows how we deal with this in our program:

```
FIELD #1,12 as NAM$,3 as Num$, 5 as Div$, 7 as Indat$, 5 as Now$.
```

Field #1 speaks for itself: 12 as NAM\$ indicates a field twelve bytes long which will take a string of that length and no longer. Similarly, the others take strings of varying lengths. To provide much flexibility, you could make each field 250 bytes long. This would be a shocking waste of memory, however, so we use the method shown. Adding up the numbers gives us a total record length of 37, which is the number you see in the program.

It should be obvious that what we have here is a means of building files for practically anything we can think of: mailing lists,



Here's an original program to organize your investments, music library, mailing lists and more!

club rosters, tape & disc libraries, recipes, library lists, and more. You name it, and this program can do it. Very little in the way of change will be required. NAM\$ can stay since it represents "name"; Num\$ also, as it represents "number". The others can be used, changed, or discarded, as you see fit.

Under the heading ReadData:, a short loop reads the record numbers from the file into variable x. The EOF(1) simply tells the program that we have reached the end of readable material in the file. Then the next few lines set various items on the screen, starting with INPUT "Stock ":NamStock\$. This will accept the name you type in for the stock, and so on down the list. Then, to enable us to build a file in the buffer to accept these entries, we use LSET NAM\$=NamStock\$. To place the contents of that buffer onto the disc we use PUT #1,x (x being the current record number we are working with).

If you goof on entering, there is a means of recovery: simply reduce x by -1 (since the process of PUTting increments the count to the next record number), and go back and enter the items again. This also gives us a means of wiping out complete files, by LSETting a series of NUL entries.

The Show: subprogram simply reads files and places everything on the screen in proper order. You will note that a few calculations are used here. I have noted these so you will know what you are dealing with.

There are three subprograms at the end of Show: which require some explanation. Use chms: to build a rectangle at the bottom of the screen which takes the two lines of text shown. A mouse routine and two lines of IF represent the limits of that rectangle in which the mouse will be effective. The third is the scr1: loop, which SCROLLS the screen one line. The -1 at the end of the SCROLL command ensures that the screen will scroll upwards; a positive number here would scroll the screen down. The SCROLL

command looks very much like the line command which we use to form a square on the screen. The numbers used represent the limits of the section of screen that will be scrolled. SCROLL is a very versatile command. It can be used to scroll the screens, or portions of it, not only up and down, but also left and right, and even diagonally.

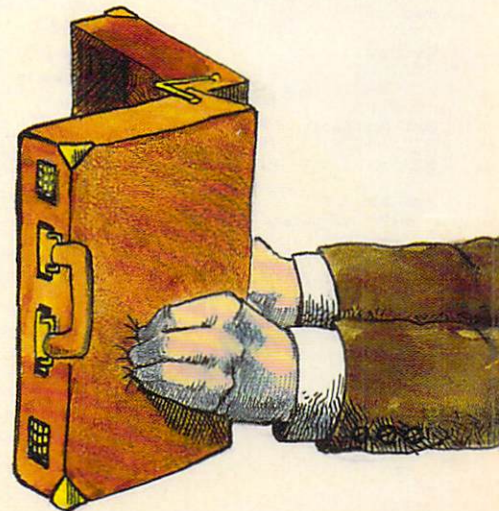
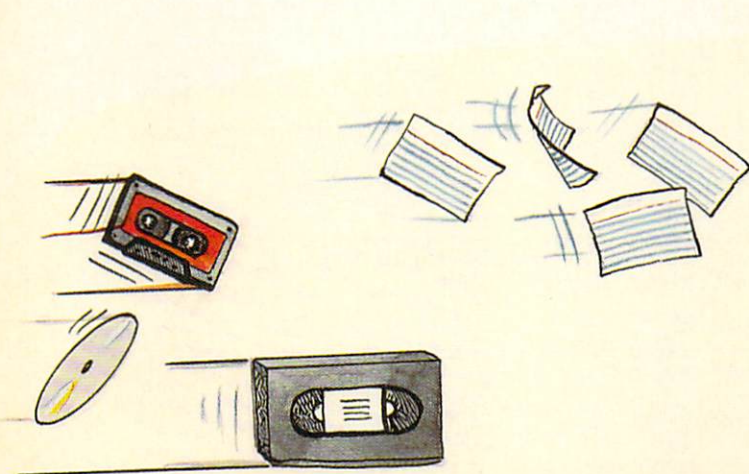
The SELL: subprogram is similar to the BUY:, but the LSET uses blank strings, as explained above, to fill the buffer before the PUT #1,x. This removes the item by overwriting it with a blank field.

When it came time to devise the subprogram to allow existing items to be changed or erased, a lot of head scratching took place! In 64/128 BASIC, you would simply set up a loop to count an array, such as FOR I=1 to 12: x(i)=a(i), and so on. I found this did not work in AmigaBASIC, and knowing so little about the use of the language by the Amiga, I was in somewhat of a quandary at the time (recently, a method of using this technique has been developed; I hope to detail it in a future program). My solution was to transpose NAM\$=NameStock\$, etc., so that it set up a buffer to accept the input. Then, when the changes were made, the whole lot was transposed again, and it worked just fine.

The Printit: sub does just that, prints it out to paper. If you look carefully at the show: and printit:, you will see some similarities. In both we invoke the PRINT, or LPRINT USING x\$="\$####.##" formula to enable our items to be printed in the spaces we need.

One point needs mentioning here. If you run out of ideas in trying to come up with names for your subprograms, AmigaBASIC will accept Line Numbers. They must be consecutive, and although I have never found the need to use them (fertile imagination, I guess!), I would suspect that they would have to follow a logical sequence with the named subs—else dire things could happen as a result.

(Listing follows)



Stock-Portfolio Listing

```
CLS
WINDOW 2,"PORTFOLIO",,15
In$=" Welcome to PORTFOLIO. "
COLOR 3,1
In=LEN(In$)
LOCATE 1,(40-(In/2))
PRINT In$
LOCATE 2,(40-(In/2))
PRINT " "
COLOR 1,0

LOCATE 8,15
PRINT "TO START PROGRAM,"
LOCATE 10,15
PRINT "TYPE FILENAME TO USE,"
LOCATE 12,15
LINE INPUT "WITHOUT EXTENSION. :";FLNS

NM$=FLNS+".REL"
WINDOW 2
GOSUB SCREENCLEAR
setscr:
LOCATE 1,20
PRINT SPACES(35)
LOCATE 2,20
PRINT SPACES(35)
LINE(25,20)-(605,175),,B
LINE(15,15)-(615,180),,B
PAINT (23,19),3,1
LOCATE 1,35
PRINT "PORTFOLIO."
LOCATE 12,23:COLOR 1,0:PRINT"MAKE YOUR CHOICE FROM THE STOCKS MENU."
:COLOR 0,1

start:
RESTORE
FOR I=1 TO 6
READ a$(I)
NEXT I
DATA "BUY STOCKS. ", "SELL STOCKS. "
DATA "SHOW STOCKS. ", "CHANGE STOCKS. "
DATA "PRINTER. ", "QUIT. "

MENU 8,0,1, "STOCKS."
FOR I=1 TO 7 : X = a(I)
MENU 8,1,1,a$(1)
MENU 8,2,1,a$(2)
MENU 8,3,1,a$(3)
MENU 8,4,1,a$(4)
MENU 8,5,1,a$(5)
MENU 8,6,1,a$(6)
NEXT I
ON MENU GOSUB selection
MENU ON
DELAY:
GOTO DELAY

selection:
ON MENU(1) GOSUB buy,sell,Show,change,printit,quit

RETURN
buy:
GOSUB SCREENCLEAR
OPEN "r",#1,NM$,37
FIELD #1,12 AS NAM$,3 AS num$,5 AS div$,7 AS Indat$,5 AS Init$,5 AS now$
now$
X=1
ReadData:
GET #1,X
IF EOF(1) THEN PRINT :GOTO Entry
X=X+1
GOTO ReadData

Entry:
LOCATE 4,8
PRINT SPACES(60)
LOCATE 4,31
COLOR 1,3 :PRINT "ENTER NEW STOCKS." :COLOR 1,0
LOCATE 6,8 :PRINT "current record No is: ";X-1
FOR I=8 TO 13
LOCATE I,16
PRINT SPACES(50)
NEXT I
FOR I=14 TO 18
LOCATE I,8 :PRINT SPACES(45)
NEXT I

LOCATE 8,8
INPUT "Stock ";NamStock$
```

```
LOCATE 9,8
INPUT "Number ";NumStock$
LOCATE 10,8
INPUT "Dividend";DivShare$
LOCATE 11,8
INPUT "Date ";DateBought$
LOCATE 12,8
INPUT "Cost ";InitCost$
LOCATE 13,8
INPUT "Current ";NowCost$
LSET NAM$=NamStock$
LSET num$=NumStock$
LSET div$=DivShare$
LSET Indat$=DateBought$
LSET Init$=InitCost$
LSET now$=NowCost$
PUT #1,X
X=X+1
LOCATE 15,8
PRINT "record "X-1;NAM$;" stored."
LOCATE 16,8
COLOR 1,3 :PRINT "IS THIS CORRECT?";
INPUT "Y/N:";ANS$
COLOR 1,0
IF UCASE$(ANS$)="N" THEN X=X-1 :GOTO Entry
LOCATE 18,8
PRINT "Add more records?";
INPUT "Y/N:";Ans2$
IF UCASE$(Ans2$)="Y" THEN GOTO Entry

CLOSE 1
GOSUB SCREENCLEAR
LOCATE 6,19
PRINT "File closed."
LOCATE 8,19
PRINT "QUIT OR RETURN TO MENU."
WICH:
a$=INKEY$:IF a$="" THEN WICH
IF UCASE$(a$)="M" THEN GOSUB SCREENCLEAR
GOSUB menuchoice :GOTO start
IF UCASE$(a$)="Q" THEN WINDOW 1:CLS :END

Show:
GOSUB SCREENCLEAR
a$="#####.##":B$="#####.##":c$="#####.##"
co=0:col=0:co2=0:co3=0:y=0:z=0:D=0:da=0
OPEN "r",#1,NM$,37
FIELD #1,12 AS NAM$,3 AS num$,5 AS div$,7 AS Indat$,5 AS Init$,5 AS now$
X=1
ReadData:
GET #1,X
IF EOF(1) THEN GOTO SHOW2
X=X+1
GOTO ReadData

SHOW2:
LOCATE 4,4
PRINT TAB(5)DATES;SPC(3)"No "SPC(2)"Divi-"SPC(2)"Pay-
"SPC(5)"Orig"SPC(3)"New "SPC(3)"RECORDS =: ";X-1
PRINT
TAB(5)"Stock"SPC(8)"Shares"SPC(2)"dend"SPC(3)"Date"SPC(5)"Cost"SPC(3)"Price
"SPC(1)"Total"SPC(6)"Gain"
PRINT TAB(5)" "
LINE (438,22)-(562,32),,B

FOR I=1 TO X-1
GET #1,I
IF I>12 THEN GOSUB chms
REP:
co=VAL(num$)*VAL(Init$) 'No of shares times initial price.
col=VAL(num$)*VAL(now$) 'No of shares times current price.
co2=col-co 'Capital gain.
co3=VAL(div$)*VAL(num$) 'Dividends payable per annum.
da=da+co3 'Total dividends per annum.
y=y+co2 'Total capital gain.
z=z+col 'Total current value.
D=da/12 'Dividends payable per month.

PRINT TAB(5)NAM$SPC(4)num$SPC(2)div$SPC(2)Indat$;
PRINT SPC(2)Init$SPC(2)now$;PRINT SPC(2)USING a$;col;
PRINT SPC(3) USING B$;co2

NEXT I
COLOR 1,0
LINE (28,148)-(603,174),,BF
COLOR 0,1
LOCATE 20,5
PRINT "TOTAL VALUE OF STOCKS: "; USING c$;z
LOCATE 22,5 :PRINT "GAIN OVER LAST ENTRY: "; USING c$;y
LOCATE 20,47
PRINT "ANNUAL DIVIDENDS : ";USING a$;da
```



```

LOCATE 22,47
PRINT "MONTHLY DIVIDENDS :";USING a$;D
CLOSE 1
LOCATE 4,4 :GOTO start

chms:
LINE (28,148)-(603,174),,B

LOCATE 20,22:PRINT "Click Left Button here to scroll."
LOCATE 21,22:PRINT "Hold button for continuous scroll."

chms1:
MOUSE ON
IF MOUSE(0)=0 GOTO chms1
X=MOUSE(1) :Y=MOUSE(2)
IF X>29 AND X<602 THEN
IF Y>145 AND Y<176 THEN GOTO scr1
GOTO chms1
END IF
scr1:
FOR S=0 TO 7
SCROLL (28,47)-(602,147),0,-1
NEXT S
LOCATE 17,6

RETURN

sell:
COLOR 0,0
LINE (27,22)-(603,175),,BF
COLOR 1,0

OPEN "r",#1,NM$,37
FIELD #1,12 AS NAM$,3 AS num$,5 AS div$,7 AS Indat$,5 AS Init$,5 AS
now$
X=1
LOCATE 4,8
PRINT SPACES(60)
LOCATE 4,33
PRINT "REMOVE STOCKS."
COUNTIT:
GET #1,X
IF EOF(1) THEN GOTO SLIST
X=X+1
GOTO COUNTIT

SLIST:
FOR I=1 TO X
GET #1,I
LOCATE 5+I,8
PRINT I;" "; NAM$
IF I=12 THEN I=X :GOTO ASKIT
NEXT I

ASKIT:
a=X
LOCATE 21,8 :PRINT SPACES(30)
LOCATE 19,8 :PRINT SPACES(35)
LOCATE 19,8 :PRINT " REMOVE RECORD (Y/N)"

askit2:
c$=INKEY$
IF UCASE$(c$)=" " THEN GOTO askit2
IF UCASE$(c$)=CHR$(13) OR UCASE$(c$)="N" THEN GOTO done
IF UCASE$(c$)="Y" THEN
LOCATE 21,8 :INPUT " RECORD NO ";n$
END IF

rmv:
LSET NAM$=""
LSET num$=""
LSET div$=""
LSET Indat$=""
LSET Init$=""
LSET now$=""

PUT #1,VAL(n$)
LOCATE 19,8 :PRINT SPACES(32)
LOCATE 19,8
PRINT "Any more to remove? ";

INPUT "Y/N";Ans2$
IF UCASE$(Ans2$)="Y" THEN
LOCATE 21,8
PRINT SPACES(32)
GOTO ASKIT
END IF

done:
COLOR 0,0

```

The Krueger Company

Processors

12 MHZ	68020	\$25.00
25 MHZ	68020	\$65.00
33 MHZ	68020	\$70.00
16 MHZ	68030	\$40.00
20 MHZ	68030	\$70.00
12 MHZ	68881	\$25.00
20 MHZ	68881	\$55.00

DRAMS

256Kx4-80 DIP/ZIP
256KX4-10 DIP/ZIP

Unconditional 30 Day Guarantee

(800) 245-2235 (602) 820-5330

Circle 116 on Reader Service card.

```

LINE (27,22)-(603,175),,BF
COLOR 0,1
CLOSE 1
GOSUB SCREENCLEAR
GOSUB menuchoice
RETURN

change:
GOSUB SCREENCLEAR
OPEN "r",#1,NM$,37
FIELD #1,12 AS NAM$,3 AS num$,5 AS div$,7 AS Indat$,5 AS Init$,5 AS
now$
X=1
LOCATE 4,8
PRINT SPACES(60)
LOCATE 4,33
COLOR 1,3 :PRINT "CHANGE ANY ITEM." :COLOR 1,0

prt3:
GET #1,X
IF EOF(1) THEN PRINT :GOTO done3
LOCATE 8,8
PRINT "Record No is: ";X

LOCATE 10,8
PRINT "1 STOCK ";NAM$
LOCATE 11,8
PRINT "2 SHARES ";num$
LOCATE 12,8
PRINT "3 DIVI ";div$
LOCATE 13,8
PRINT "4 DATE ";Indat$
LOCATE 14,8
PRINT "5 COST ";Init$
LOCATE 15,8
PRINT "6 NOW ";now$
LOCATE 17,8
NamStock$=NAM$
NumStock$=num$
DivShare$=div$
DateBought$=Indat$
InitCost$=Init$
NowCost$=now$

COLOR 1,3 :PRINT "CHANGE WHICH ITEM? ";:COLOR 1,0

```


SO YA WANNA WORK WITH VIDEO... YOU CAN DO IT!

YOU HAVE READ ABOUT IT MONTH
AFTER MONTH, BUT WHAT DOES IT
REALLY TAKE TO GET INTO THE
VIDEO FIELD WITH YOUR AMIGA?

WHAT EQUIPMENT WILL YOU
NEED?

WHAT IS MY INITIAL INVESTMENT
GOING TO COST ME?

ALL OF THESE QUESTIONS ARE
ANSWERED SO YOU KNOW
ABSOLUTELY EVERYTHING YOU
NEED TO KNOW TO GET STARTED
AND GET JOBS!

THIS LENGTHY INSTRUCTIONAL VIDEO TELLS
YOU WHICH SOFTWARE WORKS THE BEST.
WHAT HARDWARE IS NECESSARY, AND MUST OF
ALL HOW TO PRACTICALLY USE YOUR
EQUIPMENT.

COMPILED BY A WORKING PROFESSIONAL
VIDEO TEAM, THIS TAPE SHOWS YOU THE BEST
TECHNIQUES FROM SHOOTING TIPS, TO VIDEO
TITLING. MOST IMPORTANTLY WE SHOW YOU
HOW TO MARKET YOURSELF SO YOU CAN BE
PROFITABLE.

**INFO-PACKED VIDEO
FOR PROFESSIONAL
VIDEO WITH YOUR
AMIGA**

ONLY \$19.95 EACH \$34.99 FOR BOTH

Add \$2.50 per video for Shipping & Handling

CA Residents Please Add 2% sales tax per video.

Send Check Or Money Order To:

MICHAELANGELO PRODUCTIONS

1755 EL CERRITO PL. #403

L.A., CA. 90028

WE GLADLY ACCEPT VISA & MASTERCARD

Call in your order - (213) 874-7404

You can FAX us at - (213) 874-9460

ATTENTION DIGITIZERS

FINALLY A LOW COST INSTRUCTIONAL
VIDEO FOR YOUR DIGITIZER COVERING
EVERYTHING FROM CAMERA AND
LIGHTING PLACEMENT TO USING YOUR
FINISHED PRODUCT WITH YOUR
FAVORITE PAINT PROGRAM.

GET THE RESULTS YOU WANT!
MAKE MONEY WITH YOUR
DIGITIZER

LEARN TO IMPORT YOUR DIGITIZATIONS
INTO 2-D AND 3-D ANIMATION
PROGRAMS AND REALIZE THE FULL
POTENTIAL OF YOUR AMIGA!

ACHIEVE PROFESSIONAL RESULTS
USING THE KEY TECHNIQUES NOT
EVEN COVERED IN THE MANUAL!!

SECRET SOFTWARE TIPS PRECIOUSLY
UNMENTIONED WILL BRING YOUR
DIGITIZATIONS LIGHT YEARS AHEAD!!!

YOU CAN USE YOUR DIGITIZER TO
MAKE MONEY. ALL METHODS ARE
DISCUSSED IN DETAIL SO YOU CAN GET
STARTED!!!

**INFO-PACKED VIDEO
FOR PROFESSIONAL
DIGITIZING**

**We Also Do Quality
Customized
Digitizing
\$2.00 Per Picture
IFF Compatible**

Send pictures to be digitized to
the address on the left or call
us!

Circle 127 on Reader Service card.

ask2:

```
LOCATE 7,40 :PRINT " "
LOCATE 6,6
PRINT "RETURN TO CHANGE FIELD."
LOCATE 7,6
INPUT "Re-enter item THEN RETURN.(1 - 6)";ANS3$
```

```
IF ANS3$="1" THEN LOCATE 10,17 :PRINT SPACE$(19) :LOCATE 10,17
:INPUT ;NamStock$
IF ANS3$="2" THEN LOCATE 11,17 :PRINT SPACE$(19) :LOCATE 11,17
:INPUT ;NumStock$
IF ANS3$="3" THEN LOCATE 12,17 :PRINT SPACE$(19) :LOCATE 12,17 :INPUT
;DivShare$
IF ANS3$="4" THEN LOCATE 13,17 :PRINT SPACE$(19) :LOCATE 13,17 :INPUT
;DateBought$
IF ANS3$="5" THEN LOCATE 14,17 :PRINT SPACE$(19) :LOCATE 14,17 :INPUT
;InitCost$
IF ANS3$="6" THEN LOCATE 15,17 :PRINT SPACE$(19) :LOCATE 15,17 :INPUT
;NowCost$
IF UCASE$(ANS3$)=CHR$(13) THEN GOTO ASK3
```

```
ASK3:
LSET NAM$=NamStock$
LSET num$=NumStock$
LSET div$=DivShare$
LSET Indat$=DateBought$
LSET Init$=InitCost$
LSET now$=NowCost$
PUT #1,X
```

```
repl:
LOCATE 19,49 :PRINT " "
LOCATE 19,18 :PRINT "NEXT ";
COLOR 1,3:PRINT "I";:COLOR 1,0:PRINT "TEM ";
COLOR 1,3:PRINT "F";:COLOR 1,0:PRINT "IELD ";
COLOR 1,3:PRINT "E";:COLOR 1,0:PRINT "ND ";
INPUT "( I/F/E ) ";ANS3$
```

```
IF UCASE$(ANS3$)="I" THEN GOTO ask2
IF UCASE$(ANS3$)="F" THEN GOTO count3
IF UCASE$(ANS3$)="E" THEN GOTO done3
count3:
X=X+1
FOR I=10 TO 16
```

```
LOCATE 1,6
PRINT SPACE$(30)
NEXT I
GOTO prt3
```

```
done3:
GOSUB SCREENCLEAR
GOSUB menuchoice
CLOSE 1
RETURN
```

```
SCREENCLEAR:
COLOR 0,0
LINE (27,22)-(603,174),,BF
COLOR 1,0
RETURN
```

```
menuchoice:
LOCATE 12,30 :COLOR 1,3 :PRINT "NEXT MENU CHOICE." :COLOR 1,0
RETURN
```

```
printit:
GOSUB SCREENCLEAR
E$="PLEASE SET PAPER AT PERFORATION."
F$="TURN ON PRINTER TO START."
G$="NOW PRINTING."
H$="PRESS ANY KEY TO START PRINTER."
```

```
LOCATE 10,81/2-LEN(E$)/2 :PRINT E$
LOCATE 12,81/2-LEN(F$)/2 :PRINT F$
LOCATE 14,81/2-LEN(H$)/2 :PRINT H$
```

```
ASK4:
a$=INKEY$:IF a$="" THEN ASK4
LOCATE 16,81/2-LEN(G$)/2 :PRINT G$ :COLOR 0,1
```

```
OPEN "r",#1,NM$,37
FIELD #1,12 AS NAM$,3 AS num$,5 AS div$,7 AS Indat$,5 AS Init$,5 AS now$
X=1
ReadData2:
GET #1,X
X=X+1

IF EOF(1) THEN PRINT :GOTO show3
GOTO ReadData2
```

```
show3:
LPRINT TAB(5) " "
LPRINT TAB(5) DATES;SPC(3) "No. " SPC(2) "Divi-
" SPC(2) "Purch" SPC(4) "Orig" SPC(3) "New "
LPRINT
TAB(5) "Stock" SPC(8) "Shares" SPC(2) "dend" SPC(3) "Date" SPC(5) "Cost" SPC(3) "Price
" SPC(1) "Total" SPC(6) "Gain"
LPRINT TAB(5) " "
a$="#####.###":B$="#####.###":c$="#####.###"
```

```
co=0:col=0:co2=0:co3=0:y=0:z=0:D=0:dm=0
```

```
FOR I=1 TO X
GET #1,I
```

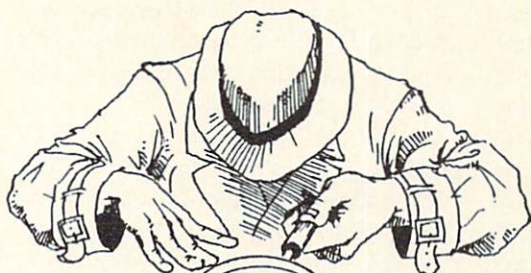
```
co=VAL(num$)*VAL(Init$) 'No of shares times initial price.
col=VAL(num$)*VAL(now$) 'No of shares times current price.
co2=col-co 'Capital gain.
co3=VAL(div$)*VAL(num$) 'Dividends payable per annum.
D=D+co3 'Total dividends per annum.
y=y+co2 'Total capital gain.
z=z+col 'Total current value.
dm=D/12 'Dividends payable per month.
```

```
LPRINT TAB(5) NAM$ SPC(4) num$ SPC(2) div$;
LPRINT SPC(2) Indat$ SPC(2) Init$ SPC(2) now$;
LPRINT SPC(2) USING a$;col;:LPRINT SPC(2) USING a$;co2
```

```
NEXT I
LPRINT :LPRINT :LPRINT
LPRINT SPC(4) "TOTAL VALUE OF STOCKS: "; USING c$;z;
LPRINT SPC(10) "ANNUAL DIVIDENDS : ";USING a$;D
LPRINT
LPRINT SPC(4) "GAIN OVER LAST ENTRY: "; USING c$;y;
LPRINT SPC(10) "MONTHLY DIVIDENDS : ";USING a$;dm
LPRINT:LPRINT:LPRINT
CLOSE 1
```

```
GOTO start
```

```
quit:
CLOSE 1
WINDOW 1
COLOR 1,0
CLS
END
```

R O O M E R S

by The Bandito

[The statements and projections presented in "Roomers" are rumors in the purest sense. The bits of information are gathered by a third-party source from whispers inside the industry. At press time, they remain unconfirmed and are printed for entertainment value only. Accordingly, the staff and associates of Amazing Computing™ cannot be held responsible for the reports made in this column.]

THAT'S ENTERTAINMENT

Is Electronic Arts having a little trouble selling cartridges? The Bandito hears that they got many unhappy returns on one of their early shipments. Seems Nintendo is in the process of developing something similar to EA's offering, and the dealers decided they wanted Nintendo's cartridge rather than the one from Electronic Arts. Whoops! Maybe the cartridge business isn't quite as easy as they thought ...

On the other hand, there is some good news. Electronic Arts has shipped their first two Sega Genesis cartridges, Budokan and Populous. To celebrate, EA gave each employee a Genesis machine and a Populous cartridge. Well, it certainly beats the usual T-shirt, that's for sure. It seems that they'll be developing a number of new titles on the Amiga (mostly in Europe) and bringing them to the Genesis.

Elsewhere in the fun-filled world of entertainment software, Mediagenic is

falling on hard times after all the recent troubles (detailed here recently). Speculation is that they may be sold at a rock-bottom price. Will Electronic Arts seize the opportunity to eliminate a rival and expand their product line of cartridges in one swell foop? Don't be surprised if it happens. Or, the buyer may be Software Toolworks, which is flush with cash after an \$80 million public offering (Software Toolworks recently made the Inc. 100 list of the fastest-growing companies).

CARTRIDGE WARS REVISITED

Nintendo will release its 16-bit Super Famicom in Japan this November, thus paving the way for an American intro in 1991. The Super Famicom uses the 65816 CPU that can handle a game cartridge with as much as 12 megabytes of memory (current carts run about 512K at the most). The screen resolution is twice that of the Nintendo, and it plays stereo sound. Of course, it won't handle standard Nintendo carts, so you'll have to buy all new carts at about \$50 a pop. The machine should sell for around \$175 dollars in the U.S., or possibly less. Apparently, Nintendo has finally felt the heat from the NEC and Sega machines.

WEST CHESTER BULLETIN

The changes in Europe may affect our friend the Amiga. According to informed sources, Commodore expects to sell 350,000 Amigas in reunited Germany

between June 1990 and January 1991. The East Germans are expected to buy huge numbers of Amigas with their saved-up ostmarks (which were converted to West German marks on July 1). At least, that's the theory. The Amiga is surely selling better than ever in Germany. Copperman will try to duplicate the German success story over here. Does that mean we'll see Amigas being sold in supermarkets? Stranger things have happened.

Commodore is also moving aggressively to capitalize on events in the rest of Eastern Europe. They plan to sell Amigas and C64's in those countries. The C64 is the perfect machine to market there, since it can be sold for less than any other computer. Commodore is working on plans to customize the C64 for the various countries over there; Poland is a prime target, as is the Soviet Union.

Developers are breathing easier over Commodore's cancellation of the C65, their new model of the C64. "It's finally dead," sighed one developer with relief. "That's one less mistake we have to worry about."

The C65 could have confused the marketplace and made it difficult to sell Amigas. It's also reassuring to note that Commodore isn't repeating the mistakes of the past. Say, maybe they could sell the C65 technology to Atari. They could call it the STjr, put a chiclet keyboard on it, and sell it as the ultimate home computer ... (!). Nah. Nobody would be that stupid, would they?

And whatever happened to the Amiga 250, the game console? Well, the prototype has been put in mothballs for now. The current thinking is to keep moving the Amiga 500 price point downwards through the mass market channels, hoping to repeat the C64 phenomenon. That is, get the price down to where sales shoot up. The A500 will be the game console, if the price gets low enough. The Bandito thinks this is a fine idea. After all, 3.5" disks are just as easy to use as cartridges, with the advantage that a game can be on more than one disk. And you get a keyboard, to boot. Leave the game consoles to Nintendo, sez The Bandito.

The Amiga 500 Professional is a sign of Commodore's new marketing thrust. Here's how it breaks down: the Amiga 500 goes into the mass-market channels and ends up being heavily discounted by Christmas (look for a \$399 street price). To help the dealers, they will offer the Amiga 500 Professional, with 1 megabyte of RAM (the daughtercard is factory installed). Funny thing—the motherboard has room for 1 megabyte of RAM, but Commodore installs the daughtercard instead, which has to cost more. Why?

So, the latest strategy is to keep dealers happy with the A500 Professional, the A2000 and 2500, and the A3000, and attack the consumer market with CDTV. This time, it looks like Commodore may be able

to pull it off. With IBM introducing a surefire flop for a home computer and Apple not far behind, 1991 could be a very good year for Commodore.

Will we see a new round of TV commercials this Christmas after the less-than-enthusiastic response last year? The word is yes, according to The Bandito's sources. With Commodore trying to move units through the mass market stores, they can't afford not to.

While we're talking about hardware, the Bridgeboard AT, with its 8 MHz 80286 CPU, is looking pretty pathetic these days (we won't even think about the original Bridgeboard). But now that the Amiga is establishing itself as its own machine, the Bridgeboard is less and less important to CBM's marketing strategy. It's possible we may see the 386 Bridgeboard, but The Bandito thinks we won't. Hey, if you want IBM compatibility, buy a cheap clone and hide it under the desk, OK?

A portable Amiga prototype is under development at Commodore, but there are no plans to bring it to market right now. The Bandito thinks it would be more sensible to modify the A500 to make it easier to carry around, and perhaps attach an LCD screen. That idea has been kicked around before, and perhaps it will get somewhere in the next year. The word is that Commodore plans to be more aggressive about developing and introducing new hardware in the future.

While you're adding up the advantages of your favorite computer, don't forget to include this one: the Amiga is the only computer that can run AmigaDOS, MS-DOS, Macintosh, and UNIX. And there's the C64 emulator and the fabled Atari ST emulator, as well. Now all we need is an Apple II emulator, and all the bases are covered ... but the best games are still on the Amiga.

Don't hold your breath waiting for KickStart 2.0 in ROM. It may not be until January that you can upgrade your A2000 with the new Denise and the new ROMs.

The A4000 is shaping up to be a killer machine for late 1991, according to data gleaned by The Bandito. Currently, it's designed to occupy an A3000 case, but with a 50 MHz 68040 on the motherboard. This

baby should really scream! Also planned is a revision of the classic Amiga graphics to allow true 24-bit color with a specialized graphics accelerator (not the normal blitter).

Hey Commodore, while you're on a roll, The Bandito has some suggestions for you. All new Amigas should have a 31.5 kHz video port and a SCSI port. And while you're at it, how about a 68030 version of the Amiga 500—a really high-powered, compact machine. And be sure to provide a slick expansion chassis for the A500 that provides at least 3 card slots. California Access showed off their version at AmiEXPO, and it looked rather slick: 4 Zorro slots, room for a hard drive and memory expansion.

Let's take a look at what's happening with Commodore's new monitors. Of course, you've already heard about the 1950, their multiscan monitor introduced with the A3000. But Commodore will also introduce the 1930, a VGA-only monitor with sound ports on the side, in the near future. This plugs into the 31.5 KHz port on the A3000 and works fine in all modes; it's cheaper than the 1950 and has a finer dot pitch. So why buy a 1950? Well, for those A500/2000 owners who have the new Denise chip, multiscan monitors are required to access the lower 15 KHz signal (which includes all the old modes and 1280 x 200/1280 x 400) video out. If you have an A3000, the 1930 looks like a better choice than the 1950.

While AmigaVision is drawing good reviews from the press and industry analysts, it's not going over so well with other developers of multimedia software. They are understandably steamed at Commodore for getting into the software business and competing against them. Look for some revisions to other multimedia products to give them a clear competitive edge over AmigaVision.

Speaking of multimedia, will Showmaker be a bust? The price is rumored to be \$395, which is very high in the Amiga market. And with a free AmigaVision ... one has to wonder. Still, Gold Disk seems quite confident. We'll just have to wait and see how many high rollers are waiting for expensive software.

Memory Management Amiga Service Specialists

Over three years experience!
Commodore authorized full service
center. Low flat rate plus parts.

Proudly affiliated with . . .

The Memory Location
396 Washington Street
Wellesley, MA 02181
(617) 237-6846

Circle 186 on Reader Service card.

Call Our FREE Order Line 1-800-992-GRYP

Digest Video Series

Tape 1 - Mastering Workbench⁺ and CLI⁺
 Step-by-step guide. Get the most from Workbench and CLI environments. **FREE** PD software and command summary. 60 minutes.

Tape 2 - DeskTop Publishing with PageStream⁺⁺
 Complete instruction - start to finish on two projects. **FREE** fonts/clipart and keyboard summary. 120 minutes.

\$30 each or both for \$50
 Includes UPS shipping - VA res. add 4.5% tax
 Call for **FREE** discount coupon and Product Guide
 Soon to be released! "Inside Workbench 2.0" and "The Power of AmigaVision"
 Grass Roots Video Productions
 P.O. Box 10889
 Burke, VA 22015
 Inquiries: (703) 569-2652
 MasterCard, VISA, Check, MO, COD
Trademark of Commodore Business Machines
 Trademark of Borland Publishing Corporation

Circle 112 on Reader Service card.

TOASTER TIMES

Of course, the hottest news in town is still the Video Toaster. The Bandito hears that some units should ship by the time you read this, though of course nowhere near enough to satisfy the demand. Elan worked on the real-time effects software package, and we can expect more cooperation between the two companies in the future. Think of what Elan Performer could be like using a Video Toaster.

HOME COMPUTERS REVISITED

Meanwhile, Apple is introducing the Macintosh Classic, the same old stuff in a new case at a slightly lower price (supposedly, Apple coughed up a million bucks for the rights to the name). Haven't any of these guys been out in the real world lately? An A500 with an AMax is cheaper than this puppy, and has color to boot. The Bandito thinks that Apple will keep trying with this, unlike IBM, because Apple just doesn't have anything else to fall back on. They have to sell a cheap Mac to get their unit sales up. But they may have to keep revising it to get it right.

In other world-shaking news, Atari is set to release the Atari TT (their 68030 machine) in this country very soon. Sales may well reach the double digits by the end of the year. The Bandito calls it the Stealth computer, in reference to all the press

coverage it has received. A bit expensive for a doorstop, don't you think? Perhaps it's just right for someone on a military budget.

BREAKING THE SPEED BARRIER

The Motorola 68040 chip release has been pushed back to October. Expect the first Amiga boards before Christmas. Watch those 3-D renderings appear in a few seconds instead of a few hours. The speed of these boards will blow you away. That is, if the price tag doesn't slay you first. Expect to pony up at least \$2,000 to be the first on your block with a 68040 Amiga.

CD-I, CDTV, CD CONFUSION DEPT.

Motorola and Philips announce that

Buy 15 1MB X 1 chips, get 1 FREE!

DRAM 1MB X 1-80	\$9.59
DRAM 256K X 4-80	\$9.95
SIMM 1MB X 8	\$95.00
ZIP 256K X 4	\$9.95

Shipping add \$3.00 /C.O.D. add \$4.00
 California residents add 6.75% sales tax

AMCS
 BOX 6981
 Salinas, CA 93912-6981

Call (408)443-1227

Circle 106 on Reader Service card.

the CD-I chip will be available next year, coinciding with the release of the hardware. Full motion video decompression on the fly will be part of the chip, although of course you'll have to compress things ahead of time (using a mondo computer and plenty of time).

How are CD-I developers responding to CDTV? Very positively. At least, here's something they can make a product for and not have to wait forever before it comes out. And CD-I machines require very expensive development systems and licensing fees, things that don't normally make a lot of friends. After all, if you're

trying to get people to support a new set of hardware, you shouldn't charge them an arm and a leg for a development system. Not to mention the fact that the tools are in the "flint knives and bearskins" stage, as Spock called it.

And Commodore is bending over backwards to get software for CDTV. Their ace in the hole: CDTV will be upgradable to CD-I; at least, that's the plan. It's possible that CDTV may be fully CD-I compatible by Christmas 1991.

On the drawing board: how about a handheld version of CDTV? A small CD-ROM player with an LCD screen and a few keys, similar to Sony's Data Discman. Possible applications in many areas, according to the Bandito's informants. It certainly is exciting.

The Bandito wants that CD-ROM drive for ordinary Amigas soon, but it may not be available until after Christmas. Look for a retail price about \$400 or under, say those in the know.

A REAL 3-D PACKAGE

Progressive Peripherals has found a way around the shelf space problem—create a box that takes up a whole shelf for their 3-D animation software. The Bandito is certainly impressed by the packaging; it even includes a videotape. The box is a cube about 12" on a side—now that's a real 3D object for you.

•AC•

WE PUT OUT !

We can transfer to color slides/prints

- ▶ 24-Bit IFF/RGB files
- ▶ ANY Amiga displayable file
- ▶ Mac, Targa, and Gif files
- ▶ AutoCad.dxf files
- ▶ And yes, Videotape!

Other services include:

- ▶ Disks of digitized pictures
- ▶ "How-to" Videos
- ▶ Custom video/graphics services

Call now for more information and ask for your free sample!



ACTION GRAPHICS

4725 Dorsey Hall Dr., Suite A402
 Ellicott City, MD 21043 (301) 992-0749

Circle 113 on Reader Service card.

(Editorial Content, continued from page 4)

Mr. Kass said there is a great demand for the services of his small company, and a particular need for more musicians. Anyone either interested in providing their artistic talents, or who need such a source of professional Amiga graphics and animation should contact:

Curt Kass
Computimation
c/o Ontological Survey
P.O. Box 17488
Milwaukee, WI 53217
(414) 332-1818

A COMPUTER LAB FOR ART EDUCATION

In a separate announcement, the University of Wisconsin and Commodore Business Machines, Inc. are teaming with Ontological to create the Art Education Electronic Instruction Media Laboratory. The facility will be housed in the Art Education Area on the Milwaukee campus of the University to provide instruction to undergraduate, graduate, and doctoral students in the use of computers in the instruction of art at the elementary, secondary, and college/university levels.

Commodore is providing Amiga computers and peripherals, in the form of a research grant. The AEEIM Lab (pronounced "aim lab") will research the use of computers in the design, articulation, and implementation of curriculum. Anyone interested in assisting Mr. Kass in this effort by providing materials, etc., should contact him at the above address.

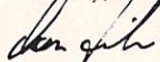
IN SUMMARY

In its use by accomplished professionals willing to teach others, and in the university and business worlds working together to create better training facilities, the Amiga has established itself as a true tool of the graphic artist. And there is a need for these talents. From product design to product marketing, the Amiga artist is necessary to provide the presentation power that business needs both in and out of the computer industry.

The need for Amiga artists was anticipated by Mr. Sachs in 1987: "Everybody's going to be starving for artists real soon." His words are stronger today than when he first said them. We need more people pushing the capabilities of the Amiga as they stand today and anticipating the possibilities of tomorrow. The Amiga offers a great opportunity, but every artist should understand the need to work hard, understand the media, and then let their imaginations soar.

As new hardware and software products become available (there are several scheduled for introduction shortly), the Amiga will maintain the edge that it has over other computer platforms. Yet, all the advances in the world will do little good without the fire and the imagination of the artist. Each of us carries the desire to produce some bit of beauty. The Amiga makes it possible. You make it happen.

Sincerely,



Don Hicks
Managing Editor

List Of Advertisers

Need more information?

Need help?

Need to know?

Contact the AC advertisers!

Please use the Reader Service Card to contact those advertisers who have sparked your interest. Advertisers want to hear from you. This is the best way they have of determining the Amiga community's interests and needs. Take a moment and contact the companies with products you want to know more about. And, if you wish to contact an advertiser directly, please tell them you saw their advertisement in

*Amazing Computing
For The Commodore Amiga*

Advertiser	Page	Reader Service Number
ACDA Corporation	29	104
AMCS	79	106
ASDG	9	108
Action Graphics	79	113
AmiEXPO	CIII	119
Beta Unlimited	71	126
Delphi Noetic Systems	54	117
Delta Graphics	71	118
Designing Minds	2	121
DigiSoft	52	137
Digital Formations	71	124
Eschalon Development	CIV	125
FD Software	45	109
G&G Technologies	12,13	122
Grapevine Group, The	34	147
Grass Roots Video Production	79	112
Great Valley Products	7	123
Hunter Group, The	65	111
InterComputing	35	114
Krueger Company, The	75	116
Memory Location, The	28	107
Memory Location, The	78	186
Michaelangelo Productions	76	127
MJ Systems	68	148
Natural Graphics	20	128
One Byte	84	135
Poor Person Software	8	129
Puzzle Factory, The	5	168
Safe Harbor Software	21	134
Silver Fox Software	14	105
TDA		
The Desktop Advantage	53	110
TRSL	22	130
Virtual Reality Laboratories	17	131

CygCC

An ARexx Programming Tutorial

by Duncan Thomson

THE LIST OF APPLICATION PROGRAMS WHICH can be controlled by ARexx, William Hawes's implementation of the REXX programming language for the Amiga, grows longer each month. Clearly, REXX has caught the attention of Amiga software developers and users, and it promises to soon become as pervasive a feature of the Amiga world as C programming and the IFF file format.

Still, despite the recent surge of interest in REXX for the Amiga, there has been a noticeable lack of really useful ARexx programs in the public domain. The examples that have appeared tend to be short and with limited function, and it is difficult to see the real power of this new language from these somewhat trivial examples. The program described in this article is intended to provide a more thorough example of an ARexx program. An example which demonstrates ARexx's remarkable power to tie together multiple application programs into one new super-program. The article will be of interest to those who just have a general interest in ARexx and its capabilities, to those who are in the process of learning to program with ARexx. As well as serving as an ARexx programming tutorial. The program presented in this article provides a function which many users may find very useful; it combines the Lattice C compiler and linker and the CygnusEd editor to provide an integrated program development environment.

The program we will be presenting is called CygCC. (The name comes from "CygnusEd editor and C Compiler".) The program provides an integrated development environment, so that after editing a C source file, the compiler and linker can be invoked with a single key press without leaving the editor. Error messages appear in an editor window,

while the lines of source code containing the errors are automatically located in the source file window. As owners of Lattice C V5.0 will know, a similar capability already exists with the Lattice Screen Editor (LSE). Why then would anyone go to the trouble of adding this capability to another editor? Simply because most programmers already like and are used to a particular editor, and are fed up with learning new ones. I myself already use four different text editors (EDT for VAX/VMS, vi for UNIX, WordStar for the IBM PC, and CygnusEd for the Amiga), and the last thing I want to do is learn another. I am sure that many other programmers out there feel the same way. Learning, yet another, new set of editor commands is not high on their list of priorities. So instead of learning to use LSE, I chose to use ARexx to add the capabilities of LSE to the CygnusEd editor. What if you do not already use CygnusEd? Then you have the chance to really learn ARexx programming by modifying CygCC so that it works with your own ARexx-compatible editor. With a bit of work, you should be able to come up with TxCC for the TxEEd editor or, as soon as the Emacs hackers build an ARexx interface into Emacs, ECC for Emacs. Similarly, if you have a C compiler other than Lattice, you are also in luck. You get to learn ARexx programming by modifying CygCC to work with your own compiler.

Before going into the details of our programming example, let us examine the features of ARexx in general. First, ARexx is a general purpose high level programming language. It has all the necessary constructs, such as IF-THEN-ELSE and DO-WHILE, to facilitate structured programming. It supports subroutines and function calls, which may even be recursive. It is implemented as an interpreter, like

BASIC, rather than as a compiler, like C. This makes program development and debugging easier at the cost of less than lightning-fast execution speed. Of course, just as compilers have been created for BASIC, a compiler version of ARexx could appear one day. However, since the most significant feature of the language is its power to control application programs. Execution speed will probably not be a critical factor for most ARexx programs, since the computationally intense grunt work can be done by the ARexx application programs. Because of this, the need for a compiler is diminished, and the interpretive approach works quite nicely.

As we have stated, ARexx's most significant feature is its ability to communicate with and to control other programs. At first, this ability seems a bit mysterious, but the implementation of this bit of magic is actually quite simple. It is made possible by the Amiga's true multitasking operating system. You are not likely to see an implementation of ARexx for MS-DOS, or even for the Macintosh - chalk up another first in the microcomputer world for the Amiga! So how does it work? First, the application program, "host" in ARexx terminology, must be designed to work with ARexx. This might seem like a serious limitation, but it is actually quite a trivial task for a software manufacturer to add an ARexx interface to their product, and the list of Amiga applications which support ARexx is constantly growing. To communicate with ARexx, the host creates a public message port which ARexx can send messages to. (Messages and message ports are features of AmigaDOS, but if you are not familiar with them, do not worry, you need not know how it works in order to write ARexx programs - this is for background information only.) When the ARexx interpreter comes across a line in an ARexx program which it does not recognize as an internal ARexx statement, it assumes that it is a command for the host. It packages the command up in a message, which is sent to the host's message port. The host treats commands received at this message port just as if they had been entered from the keyboard or with the mouse, so that the full power of the application becomes available for use in your ARexx program.

For controlling programs which do not include an ARexx interface, such as the Lattice C compiler, we have one more trick up our sleeve. That is the ability to issue DOS commands from within an ARexx program. There are some limitations on this, due to the fact that the Amiga Shell nor the CLI does not have a built-in ARexx interface. The most significant limitation is the inability of an ARexx program to obtain the result code set by a DOS command. This makes it hard to tell whether the command succeeded or failed. It is reported that in AmigaDOS 1.4, the Shell will be updated to include an ARexx interface, so that it can act as a true ARexx host. Also, a Shell replacement which includes an ARexx interface, called WShell, is currently available from William Hawes, the maker of ARexx. However, we will see in the example program that the ARexx/AmigaDOS interface works fairly well even without these solutions.

Communications between ARexx and other application programs (hosts) is two-way. In addition to accepting commands from ARexx programs, as described above, hosts can also invoke ARexx programs. This two-way communication makes it possible for ARexx to replace the macro languages that are often included in applications to allow users to customize them to meet their own needs. The user can write an ARexx program which when invoked by the host, turns around and sends back to the host a series of commands which carries out the desired function.

There are two advantages to replacing application-specific macro languages with ARexx. First, macro languages included with applications, such as editors, spreadsheets, etc., tend to be somewhat crude and limited in their programming features, while ARexx provides all the power of a structured, general purpose programming language. Second, ARexx will soon provide a common macro language which can be used with almost all Amiga applications, so that users will not have to learn a new macro language each time they acquire a new tool.

THE EXAMPLE PROGRAM

Without further ado, let us take a look at the programming example. Figure One shows the structure of our project, which consists of four modules: CygCC, ReadErrors, NextErr, and FindWind. CygCC is the module which runs the compiler and linker. When the user invokes CygCC, the source file will be saved. The compiler will then be run with this source file as input. If the compilation is successful, the linker will be run. CygCC will open up a new editor window, titled Compiler_Window, and will invoke ReadErrors to copy into this window any error or warning messages produced by the compiler and linker. After the compilation is complete, CygCC will invoke NextErr to highlight the first error message in Compiler_Window and to simultaneously locate the line containing the error in the source file window. The user can then repeatedly invoke NextErr to advance through the error messages, and to locate and correct his errors in the source file, one by one. FindWind is a little utility routine which is used by the other modules to look for a named editor window.

CygnusEd allows ARexx programs to be "bound" to function keys, so that pressing the key invokes the program. The user will find it convenient to bind CygCC to one key, and NextErr to another. In this way the compiler can be invoked with one keypress, and errors can be located and viewed with another keypress.

FIND A WINDOW - FINDWIND

Since FindWind (shown in Listing One) is the simplest of our four modules, let us tackle it first. FindWind is an ARexx function which will return a one to its caller if it can locate a specified editor window (called a "view" in CygnusEd terminology), and a zero if it cannot. Since it is our first example, we will look at it in some detail. It begins, as all ARexx modules must, with a comment. Comments in ARexx are begun with a "/*" and ended with a "*/", just as in C. The first executable statement in FindWind is the following:

```
if address() ~= 'rexex_ced' then return 0
```

This is an example of the IF statement. The purpose of this statement is to find out if the current host is the CygnusEd editor. FindWind is only designed to be called from a CygnusEd application, so if the current host is not CygnusEd, then the function will simply quit, returning a zero to indicate failure. To determine the current host, we use the built-in ARexx function address(). This function returns a string giving the name of the current host. We compare this with the host name used by CygnusEd: rexex_ced. The operator for inequality is "~=". Note that, unlike C and Pascal, ARexx allows us a great deal of flexibility in manipulating strings, including the ability to directly compare them. If the two strings are not equal, the THEN clause of the

statement is executed, causing FindWind to return to the caller with a result of 0.

Next we come to:

```
arg window_name
```

This illustrates the ARG statement, which is used to obtain the arguments which have been passed to a function. A statement of the form "arg a1 a2 ..." assigns to the variable a1, the value of the first argument, to a2, the second argument, and so on. In our case the variable window_name is now set to whatever value was given when FindWind was invoked. Note that we do not need to declare variables, we just go ahead and use them at will.

The next statement is one case of the OPTIONS statement which is used to set various options which control the execution environment. For now, all we want to do is allow the host to send back results after we send commands to it. This is done with:

```
options results
```

The next pair of statements is our first example of a host command:

```
'Status 66' /* Get number of views */  
num_views = result
```

The ARexx interpreter will not be able to recognize the command "Status 66", so it will send it off to the current host (which we know is CygnusEd). The status command is part of the CygnusEd ARexx interface that is used for getting information from the editor. Different values following the command indicate requests for different pieces of information. The value 66 indicates that we want to know the number of "views", or editor windows, in existence. Since we have enabled the return of results from the host, the result of the command will appear in a special variable called (you guessed it) result. The next statement simply saves this value in a variable called num_views. For C and Pascal programmers, creating variables like this, without having declared them first, takes a bit of getting used to, but it is actually very convenient. BASIC programmers, of course, will find this approach quite natural.

Some readers may be wondering at this point about the use of the apostrophes in the above code. In ARexx, either apostrophes or quotation marks may be used to delimit literal strings. Examples of these are 'rexex_ced' and 'Status 66'. We have used the apostrophes to make these into literal strings to prevent ARexx from attempting to evaluate them as variables.

Now we come to something a bit more interesting: a loop construct. What we want to do is loop through each of the editor windows until we find the one we want. This is accomplished with the following code:

```
/* Loop until found specified view or  
out of views */  
found = 0  
do for num_views until found  
  'Status 21' /* Get view name */  
  if upper(result)=window_name then  
    found = 1  
  else
```

```
    'Next view'  
  end
```

We are using the variable found to keep track of whether we have found the window we are looking for. In ARexx, a value of zero is interpreted as FALSE and a value of one is interpreted as TRUE, so we initialize found to zero to indicate that we have not found our window yet. The loop is accomplished with a DO statement. The DO statement is a very powerful ARexx construct that has several forms. The form we are using will cause the body of the loop, all the statements between DO and END, to be carried out num_views times, or until found becomes true. The body of the loop is easy to understand. We get the name of the current editor window using the "Status" command we saw before. If it is the name we are looking for we set found true, which will get us out of the loop. If it is not the one we want we send the "Next view" command to the editor to advance us to the next window. Note the slightly different form of the IF statement used here. This one has an ELSE clause, which works just as you would expect it to.

Now we are done with FindWind. All that remains to do is to pass back to the caller the result of the search, which is contained in the variable found:

```
return found
```

THE MAIN MODULE: CYGCC

Now we'd better dive in and look at the main body of our example program, the module CygCC (Listing Two). We begin by defining a couple of variables:

```
nl = '0A'X  
stacksize = 40000
```

The variable nl will be used to write newline characters on output. Note how we can set a variable to any arbitrary hex value by following a string of hex digits by an X.

Next we check, as we did in FindWind, to make sure that CygCC is being invoked from within the editor:

```
if address() ~='rexex_ced' then do  
  say 'This program must be run from within CygnusEd'  
  say 'Press return to continue...'  
  pull answer  
  exit  
end
```

This time, if the host is not what we expect it to be, we provide an error message to the user with the SAY command and use the PULL command to wait until the user presses the return key. Note the use of the DO - END construct here to include several statements within the THEN clause of the IF statement. In PASCAL this would have been done with a BEGIN - END construct; in C with the use of curly braces. In ARexx it is the ubiquitous DO statement which does the trick.

The next statement is one we have seen before:

```
options results
```

This is needed to allow the host to pass result strings back to our ARexx program.

We take a **byte** out of the price!



ONE BYTE

P.O. Box 455
Quaker Hill, CT 06375
(800) 441-BYTE, In CT (203) 443-4623

YOUR ONE-STOP  STORE

**Authorized dealer for
Commodore-Amiga Computers,
Great Valley Products (GVP),
Authorized Commodore-Amiga Service and Repair.
Authorized Amiga Graphics Dealer.**

AMIGA IS A REGISTERED TRADEMARK OF COMMODORE-AMIGA, INC.

Circle 135 on Reader Service card.

Preliminaries completed, we proceed to get some information from the editor with the following:

```
'Status 17'
nlines = result /* Number of lines in file */
'Status 19'
fullname = result /* Full path and name */
'Status 20'
pathname = result /* Path only */
'Status 21'
filename = result /* Name only */
```

Now we have got various forms of the name of the file that the user was editing when CygCC was invoked. The next group of statements is used to check that the user was actually editing some file, and that its name ended in ".C", as we expect from a C source file. We also perform some manipulations to strip off the ".C" extension and make sure that path ends in a ":" or a "/". The code should be self-explanatory. Note the use of some of ARexx's rich library of string manipulation routines: lastpos() to get the location of the last appearance of a given character in a string, length() to get the length of a string, right() and left() to extract ending and beginning substrings from a string, and upper() to translate a string to upper case. Also note the use of the "||" operator to concatenate strings. Strings can also be concatenated simply by writing them one after the other, but in that case a blank will be inserted between the two strings.

```
/* If filename hasn't been defined, we can not go on */
```

```
if filename='' then do
  'Okay! Source file name not defined'
  exit
end
/* Strip the .C extension and get bare file name */
barename = ''
dotpos = lastpos('.',filename)
if dotpos>0 then do
  ext = right(filename,length(filename)-dotpos)
  if upper(ext) = 'C' then
    barename = left(filename,dotpos-1)
end
/* If name didn't end in ".C" */
if barename='' then do
  'Okay! Error: Filename does not end in ".C"'
  exit
end
/* Make sure pathname ends in a ':' or a '/' */
if right(pathname,1) ~= ':' then
  pathname = pathname||'/'
```

The "Okay1" command that appears in the above code fragment is a CygnusEd command which brings up a requester containing the given message and waits for the user to click on the requester before continuing.

We have already seen how information can be passed between ARexx modules by means of argument passing. The next few lines illustrate another method of sharing information. The NextErr function will need to know how many lines were in the source file when it was compiled (we will see why later). We cannot simply pass this information to NextErr as an argument, since NextErr will be invoked as an independent program directly by the user as he steps through and corrects his errors one at a time. Our solution is to use an ARexx facility called the clip list. Information can be saved in this permanent, globally accessible storage facility by using the setclip() function. We will see later how the information is retrieved with the getclip() function.

```
/* Save the number of source lines on clip list,
for use by NextErr */
call setclip( 'CygCC_NLines', nlines )
```

Before we can compile, CygnusEd must be told to save the file to disk, where the compiler can get at it. This is easily accomplished with a single editor command:

```
'Save'
```

The next step is to create a new editor window which the compiler error messages are going to appear in. If the window already exists (from a previous compilation perhaps), then we delete it with a "Quit" command and open a new one. We then use the CygnusEd "Text" command to insert a message into the new window telling the user that we are about to compile his file.

```
if FindWind( 'Compiler_Window' ) then 'Quit 1'
'Open new'
'Open Compiler_Window'
'Text Compiling' fullname||nl
```

Some of the error messages produced by the linker contain escape codes which will mess up the CygnusEd window, so we tell CygnusEd to make escape codes invisible:


```

/* If escape codes are visible, make them invisible */
'Status 34'
if result then 'Esc codes visible' /* Toggle off */

```

Now at last we can run the compiler. The Lattice compiler actually consists of two phases, called LC1 and LC2. The first step is to run LC1, as follows:

```

address command,
  'c:stack' stacksize nl,
  'lc:lc1 > t:phase1 -oquad:' fullname

```

The above statement is worth looking at in detail. ARExx statements are normally placed on one line, but in this case we have one statement which is split into three lines. The comma at the end of a line serves as a continuation marker, telling ARExx that the statement is continued on the next line. The statement is a special form of the ADDRESS command, which has two uses: to switch between different hosts, and to send DOS commands. The keyword command tells ARExx that we are using the second form here. The keyword must be followed by a string containing the DOS command to be executed. In the statement above the command string is built by concatenating together a number of strings. (Remember that ARExx simply concatenates together strings which are written one after another.) Both literal strings, such as 'c:stack', and variables, such as stacksize, nl, and fullname, are used. Remember that nl is a variable which we created earlier to contain the new-line character, hexadecimal OA. By inserting this in the middle of a DOS command string, it is possible to issue multiple DOS commands at one time. The first command increases the stack, so that the compiler will not choke on a big source file. The second command runs the first phase of the compiler, redirecting compiler messages to the file T:PHASE1, placing the compilation output in the QUAD directory.

In your DOS startup sequence you should have T assigned to some place you use for temporary storage, such as the directory RAM:T. When you set up your C compiler, you also should have assigned LC to the directory containing the compiler, and QUAD to the directory you want to place compiler intermediate files in. Making QUAD be RAM will speed up compilation, but be careful; remember that you will have the compiler, the ARExx interpreter, and the CygnusEd editor, containing your source file, all in memory at one time. I have got a standard one megabyte A2000, and if I assign QUAD to be RAM, I will run out of memory when compiling a medium size source file. Luckily I also have a fast hard disk, so I simply assign QUAD to a directory on the hard disk, which is almost as fast as using RAM.

Once the first phase of compilation is complete, we go and read the compiler messages, from T:PHASE1, and copy them into the window we have created. This task is carried out by the routine ReadErrors. We will see how this routine works later. For now, all we need to know is that it will return the number of error messages the compiler has output. We will store the number of errors in a variable called cc_err_count.

```
cc_err_count = ReadErrors( 't:phase1', 'LC1' )
```

If no errors were detected in the first phase, we proceed to run the second phase. The code for this is similar to the code we used to run the first phase:

```

address command,
  'c:stack' stacksize nl,
  'lc:lc2 > t:phase2 quad:'||barename
cc_err_count = ReadErrors( 't:phase2', 'LC2' )

```

If no errors were detected in the second phase of compilation, we can run the linker. We get a bit tricky here - we use the ARExx built-in function exists() to look for a file with the same name as the source file and an extension of ".LNK". If it exists, we use this as the command file which contains all the options needed to run the linker. If not, we just supply a default set of options to the linker, as follows:

```

/* If a linker command file exists */
if exists( pathname||barename||'.lnk' ) then do
  /* Insert message to tell user we are linking */
  'Text Linking WITH' pathname||barename||'.lnk' ||nl
  /* Link using the command file */
  address command,
    'c:stack' stacksize nl,
    'c:cd' pathname nl,
    'lc:blink > t:link WITH',
    pathname||barename||'.lnk'
end; else do
  /* Insert message to tell user we are linking */
  'Text Linking' pathname||barename||'.o' ||nl
  /* Link using default options, libraries, etc. */
  address command,
    'c:stack' stacksize nl,
    'lc:blink > t:link FROM lib:c.o',
    'quad:'||barename||'.o TO 'pathname||barename,
    'LIBRARY lib:lc.lib lib:amiga.lib BATCH'
end
/* Read the linker error messages */
lnk_err_count = ReadErrors( 't:link', 'LINK' )

```

Now the grunt work is all done. All that remains is to show the user the results of the compilation. If we had compiler errors, the thing to do is to locate the first error in the source file, so that the user can begin to correct the errors. This is done by commanding the editor to move to the top of the compiler message window, and then calling our function NextErr, as follows:

```

/* Find the first error message */
'Beg of file'
'Mark block'
call NextErr()

```

On the other hand, if we had no compiler errors, then we were either completely successful, or we had linker errors. We issue an message telling the user what happened, and then use CygnusEd's "Okay2" command to allow the user to delete or keep the compiler window with a click of the mouse button. The code is as follows:

```

if lnk_err_count=0 then
  message = 'Success!'
else
  message = 'Link Failure...'
/* Quit, allowing user to either delete error
window or keep it */
'Okay2' message 'Delete compiler window?'
if result then
  'Quit 1'
else do
  /* Move to top of error window */
  'Beg of file'
  'Mark block'
  /* Make source file window active */

```



```
FindWind( filename )
end
```

There is only one executable statement left in CygCC, and you can guess what it does:

```
exit
```

THE ERROR READER - READERRORS

The module ReadErrors (Listing Three) is called to read the compiler or linker error messages, which have been redirected to a file, and to transfer the messages into an editor window. The messages are written in the error window in a format which will be easy to read and interpret in the module NextErr.

Most of the code in ReadErrors is quite similar to the code we have seen in the previous two modules, so rather than go through it line by line, let us just examine the most interesting points.

ReadErrors provides our first example of file I/O. The file containing the compiler messages must be opened and then read, one line at a time. To open the file, the ARexx built-in function open() is used. The first argument to open() is a logical name, which will be used to identify the file from now on. The second argument is the actual file name, and the third argument is how we want to access the file. In this case we want to read the file, so the third argument is set to Read. The open() function returns TRUE if the file was opened successfully, and FALSE otherwise. In the following code we open the file and check the result. If the result is not TRUE (the tilde means "not"), then we give an error message and abort the program.

```
/* Open input file - quit if we can not open it */
if ~open( infile, err_file, 'Read' ) then do
  'Okay! Unable to open file:' err_file
  exit
end
```

Once the file has been opened, we use the readln() function to read lines from it, and the eof() function to determine when all lines have been read from the file. You will see that the structure of ReadErrors looks like this:

```
open( infile, err_file, 'Read' )
instring = readln( infile )
do until eof( infile )
  ...process the data in variable instring...
  instring = readln( infile )
end
```

This shows how file input is accomplished. Output is accomplished in a similar fashion.

The second point of interest in ReadErrors is the use of internal functions. Let us take a closer look at how subroutines (called functions in ARexx terminology) are handled. There are two ways that functions can be called in ARexx, either explicitly, with the CALL statement, or implicitly, by simply referencing a function in an expression.

```
call mysub( args ) /* explicit subroutine call */
x = yours( args ) /* implicit call */
```

No matter how a function is called, ARexx applies the same procedure for finding it. First, ARexx looks for the subroutine name, followed by a colon, within the current file. This is called an internal function, and is illustrated in ReadErrors by the functions ReadLinkError(), ProcessFlagLine(), and ProcessOth-

erLine(). If ARexx cannot find a function this way, it checks for a built-in function. We have seen many instances of built-in functions - open(), eof(), and readln() for example. If the function is not a built-in function, ARexx checks available function libraries and function hosts, which we have no examples of in this program. Last, ARexx looks for a file on disk with the same name as the function. ReadErrors(), FindWind(), and NextErr() are all examples of this type of function.

The last point of interest in ReadErrors is the use of the PARSE command, which provides great power for breaking up strings into their component parts. This instruction has many different forms and options. One of these forms is illustrated by the code we used to break up a compiler error message, contained in the variable instring:

```
parse var instring fname line_num severity . message
```

This line of code breaks up instring into substrings using blank space as delimiters. The first substring is placed in the variable fname, the second in line_num, the third in severity, and the remainder of instring is placed in the variable message.

THE ERROR LOCATOR - NEXTERR

The last of our modules is NextErr (Listing Four), which highlights the next error message in the message window, and jumps the cursor to the line containing the error in the source file window. If you have been able to follow the explanation of the previous three modules, then you will find the code of NextErr self-explanatory. The only point worth noting is the use of the GetClip() function, which corresponds to the SetClip() function used in CygCC. We use GetClip() to recover from the clipboard the number of lines which were in the source file when it was compiled. Using this information, NextErr can correctly find the line containing the next error, even if the user has added or deleted lines in the course of correcting previous errors.

IN CONCLUSION...

The CygCC program presented in this article was intended to provide an in-depth introduction to ARexx with a really functional example program. There are several interesting aspects of ARexx that we could not address in the space of this article - compound variables, interactive tracing, and function libraries, to name just a few. Nevertheless, readers who have followed through the example code should now have a good idea of how ARexx can be used to bring several different application programs together into one coherent whole. Of course, the only way to really learn a new programming language is to begin to use it yourself. As was mentioned at the beginning of this article, if you use a different editor or programming language, a good starting ARexx project might be to adapt the CygCC program to work with your editor and language.

The CygCC program is placed in the public domain, and you are free to use it as you wish. But if you do come up with improvements or new versions of CygCC, or some new ARexx programs of your own, please consider placing them in the public domain, so that we can all use them. Since ARexx is implemented as an interpreted language, programs must be distributed in source code format. This means that there probably will not be much ARexx code written by commercial developers, so for our supply of good ARexx programs we will be relying on that great Amiga resource: shareware and public domain software.

(continued on page 88)

C NOTES

From the C Group
by Stephen Kemp

PROBABLY EVERYONE WHO PROGRAMS ON THE AMIGA, whether professional or amateur, knows that the machine is capable of multitasking. Although, this is a well-known accomplishment for the Amiga, many programmers (even professional ones) fail to consider that this also means it is possible to run the same program simultaneously—perhaps a program that you wrote. The reason I mention this fact is to illustrate the point that you will have to plan for this event while you write your C programs (as well as those in any other language).

If you are writing a program that opens a data file, have you planned on “sharing” this file, or do you expect to have exclusive access? This is an important question to ask yourself. Sharing the data file sounds simple enough, but what happens when the file is changed by your other self (or another program)? Does the possibility exist that one program can change something in the file, only to have it changed again by the other? These are important questions and will be application-dependent. If the file in question is being updated in some “batch” mode, with numerous revisions on a series of records, then you may have a potential problem. Although this is probably more likely to be a problem found in a network environment rather than multitasking, you may not be there to operate your software, and your client may not know of the potential harm of running the program again simultaneously.

What you do in a situation like this depends upon a number of factors. If the operating system offers a method of file control, then that can be used. If it is pretty much left to the programmer then you have to develop your own techniques. One simple solution involves developing your own version of file protection. With a control file (or even the first record within the data file) you can indicate if the file is in use. With this method, you would first check to see if it is okay to proceed by examining the “in-use” indicator (this could be a character, a signature word, or any specified data that you can read and examine). If access is allowed, you would change the indicator and store it back into the appropriate place so a simultaneous run of the program would result in the second program handling the situation.

Determining that the file was in use might call for waiting until the indicator is changed, or simply terminating the run. You probably should offer an override capability in the event that the indicator can be changed if a previous run did not complete successfully and left the in-use indicator on accidentally.

This technique can actually be carried even further and used to indicate specific areas within a file that are in use, rather than simply marking the entire file. Essentially, you could include indicators on each record and allow access to all of the file except the record with which your program is currently working. Many networking (and DOS) environments offer these features automatically through system calls and do not require you to make room in your data files for the indicators.

Another file event that is often overlooked by many programmers in the multitasking (and networking) environment is in creating files for outputting information, like reports. Suppose you have written a program that takes a document and builds an index on some specified key words that might be found within the text. The index file that you build has to be given a name. If you elect to use a “hard-coded” name like, INDEX.FIL, you have a potential hazard. If a subsequent run of your program begins before the first completes you will clash in the output file, probably producing incorrect results.

Usually, it is best to avoid these situations by allowing the user to specify output files when possible. It is not difficult and probably could be made into a library function that you could include in all your programs without having to duplicate the effort.

The same type of thing can occur with programs that employ temporary files. I have seen this happen on several occasions with programs doing sorts on large data files. Often it is impossible to read an entire file into memory and sort it, so temporary files are required to complete the task. You can see what will happen if the program assigns the same name (or series of names) to its temporary file(s). If another task is started with the same program then you may end up deleting or changing the temporary file(s) without the other task knowing; or having it done to your file(s) by the other task.

Temporary files are, unfortunately, a must for many programs. Since they are “temporary” files, it is probably unreasonable to expect the user to type in a name each time one is required. However, you can avoid these problems by including a few more instructions in your code. First, develop a function (or determine if there is one already in your C libraries) that will produce a file with a unique name. This usually involves some text and numbering scheme, like TMP00001.TMP. Within some loops, you would create a name (like the one above) and then

(C Notes, continued)

check to see if a file by that name already exists. If one does, then you start over and create another name (like TMP00002.TMP) and try again. Eventually, you will develop a unique name that can be created for your temporary file. (NOTE: If you do decide on this method you may want to use a random number rather than beginning sequentially as I have demonstrated. This will improve your chances of getting a unique name on the first few tries.) Remember to remove any temporary files that you create when you are finished with them. There is nothing worse than a disk full of junk.

These are some of the things that you need to consider when writing programs that do file manipulation. Of course, there are many others and I do not necessarily offer these suggestions as the best solutions. You, as the programmer, have to determine how to handle these situations in your programs. The point that I want to stress is that file relationships need to be considered during program development. I don't like it when programs I use have these problems and you probably have had the same experience. So, think about how your program will react if it "meets itself" during a run.

•AC•

(CygCC, continued from page 86)

Listing One

```
/*
 * FindWind - find a window (view) and make it active.
 *           (must be run from CygnusEd)
 *
 * Input argument: name of window
 *
 * Returns: 1 if success, 0 otherwise
 */

/* The current address must be rexx_ced */
if address() ~= 'rexx_ced' then return 0

/* Get the arguments */
arg window_name

/* Allow host to return results */
options results

'status 66' /* Get number of views */
num_views = result

/* Loop until found specified view or out of views */
found = 0
do for num_views until found
  'Status 21' /* Get view name */
  if upper(result)=window_name then
    found = 1
  else
    'Next view'
end

return found

/* End FindWind */
```

Listing Two

```
/*
 * CygCC - Run the compiler from within CygnusEd editor
 *
 * AREXX program written August 1989, by Duncan Thomson
 *
 * This is the main driver of a program which invokes
 * the Lattice C compiler and linker. It must be called
 * from the CygnusEd editor.
 *
 * The following AREXX programs are also required:
 *   FindWind - finds a particular CygnusEd window
 *   ReadErrors - copies compiler messages to window
 *   NextErr - advances to next error message and
 *             corresponding line of source code
 */

/* Define some miscellaneous variables we will need */
nl = '0A'X
stacksize = 40000

/* Make sure the current host is CygnusEd */
if address() ~= 'rexx_ced' then do
  say 'This program must be run from within CygnusEd'
  say 'Press return to continue...'
  pull answer
  exit
end

/* Allow CygnusEd to pass status variables */
options results

/* Get info about current file from CygnusEd */
'status 17'
nlines = result /* Number of lines in file */
'status 19'
fullname = result /* Full path and name */
'status 20'
pathname = result /* Path only */
'status 21'
filename = result /* Name only */

/* If filename has not been defined, we can't go on */
if filename='' then do
  'Okay! Source file name not defined'
  exit
end

/* Strip the .C extension and get the bare file name */
barename = ''
dotpos = lastpos('.',filename)
if dotpos>0 then do
  ext = right(filename,length(filename)-dotpos)
  if upper(ext) = 'C' then
    barename = left(filename,dotpos-1)
end

/* If name didn't end in ".C" */
if barename='' then do
  'Okay! Error: Filename does not end in ".C"'
  exit
end

/* Make sure pathname ends in a ':' or a '/' */
if right(pathname,1) ~= ':' then
  pathname = pathname||'/'

/* Save the number of source lines on clip list,
for use by NextErr */
call setclip('CygCC_NLines', nlines)

/* Save the file, in the current directory */
'Save'

/* Open compiler window to show messages in */
if FindWind('Compiler_Window') then 'Quit 1'
'Open new'
'Open Compiler_Window'
'Text Compiling' fullname||nl

/* If escape codes are visible, make them invisible */
'status 34'
if result then 'Esc codes visible' /* Toggle off */

/* Run the 1st phase of the C compiler */
address command,
  'c:stack' stacksize nl,
  'lc:lcl > t:phasel -oquad:' fullname
cc_err_count = ReadErrors('t:phasel', 'LCL')

/* If first phase completed with no errors */
if( cc_err_count=0 ) then do
```



```

/* Run the 2nd phase of the C compiler */
address command,
'c:stack' stacksize n1,
'lc:lc2 > t:phase2 quad:'||barename
cc_err_count = ReadErrors( 't:phase2', 'LC2' )

/* If 2nd phase completed with no errors */
if( cc_err_count=0 ) then do

/* If a linker command file exists */
if exists( pathname||barename||'.lnk' ) then do
/* Insert message to tell user we're linking */
'Text Linking WITH' pathname||barename||'.lnk' ||nl
/* Link using the command file */
address command,
'c:stack' stacksize n1,
'c:cd' pathname n1,
'lc:blink > t:link WITH',
pathname||barename||'.lnk'
end; else do
/* Insert message to tell user we're linking */
'Text Linking' pathname||barename||'.o' ||nl
/* Link using default options, libraries, etc. */
address command,
'c:stack' stacksize n1,
'lc:blink > t:link FROM lib:c.o',
'quad:'||barename||'.o TO 'pathname||barename,
'LIBRARY lib:lc.lib lib:amiga.lib BATCH'
end
/* Read the linker error messages */
lnk_err_count = ReadErrors( 't:link', 'LINK' )

end

end

/* Show user the results of the compilation */
if cc_err_count>0 then do

/* Find the first error message */
'Beg of file'
'Mark block'
call NextErr()

end; else do

if lnk_err_count=0 then
message = 'Success!'
else
message = 'Link Failure...'

/* Quit, allowing user to either delete error
window or keep it */
'Okay2' message 'Delete compiler window?'
if result then
'Quit 1'
else do
/* Move to top of error window */
'Beg of file'
'Mark block'

/* Make source file window active */
FindWind( filename )

end
end
end

```

Listing Three

```

*****
*
* ReadErrors - process the output of LC1 or LC2 (1st
* stage or 2nd stage of the Lattice C
* compiler) or the linker.
*
* To be invoked by CygCC, the CygnusEd compiler invoker.
*
* NOTE - works with Lattice C version 5
*
* Inputs:
* err_file - file containing compiler outputs
* file_type - what type of output are we reading:
* LC1, LC2, or LINK
*
* Output:
* returns the number of error messages found (not
* counting warnings)
*/

/* Allow host to return results */
options results

/* Get the arguments */
arg err_file, file_type

/* ASCII character definitions... */

```

```

nl = '0A'X /* New Line */
flag_char = '9B'X /* C error messages contain this */
esc_char = '1B'X /* Some link messages contain this */

/* Find the view to put the messages into */
if ~FindWind( 'Compiler_Window' ) then do
/* Quit if we couldn't find window */
'Okay1 Unable to find Compiler_Window'
exit
end

/* Open input file - quit if we can't open it */
if ~open( infile, err_file, 'Read' ) then do
'Okay1 Unable to open file:' err_file
exit
end

/* Read lines from input file, until none left */
line_count=0
err_count=0
column=0
instring = readln( infile )
do until eof( infile )

/* Skip first two lines of "banner" text */
line_count = line_count+1
if line_count>2 then

/* If we are reading the output of the linker */
if file_type='LINK' then
call ReadLinkError()
else do /* Else, must be compiler output */
/* If the string contains the flag character */
flag_pos = index( instring, flag_char )
if flag_pos>0 then
call ProcessFlagLine()
else
call ProcessOtherLine()
end
end

/* Get the next line of input */
instring = readln( infile )

end

/* Now get rid of the error file */
cl_res = close( infile )
address command 'Delete' err_file

return err_count
/* end of function ReadErrors */

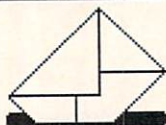
ReadLinkError:
/* If line contains an ESC character or "error" */
if index( instring, esc_char )>0 ,
| index( upper( instring ), 'ERROR' )>0 then
/* It's some kind of linker error message */
err_count = err_count+1
/* If the line is not blank */
if instring=' ' then
/* Copy it into the editor window */
'Text' instring ||nl
return
/* End of function ReadLinkError */

ProcessFlagLine:
/* If this is a system error message */
if index( instring, 'Stack Overflow' ) > 0 then do
'Text System <<< Stack Overflow >>>' * '||nl
err_count = err_count+1
end; else do
/* This is a line of the user's C source with the
error position marked by the flag character */
column = flag_pos
end
return
/* End of function ProcessFlagLine */

ProcessOtherLine:
/* If previous line contained a flag character */
if column>0 then do
/* Current line contains file name, line num, etc.*/
parse var instring fname line_num severity . message
/* Reformat error message into Cygnused window */
'Text' left( severity, 9 ) '<<<' || message '>>>',
fname line_num column ||nl
if severity='Warning' then err_count=err_count+1
column = 0
end; else do
/* Look at the first word on the line */
parse var instring firstword .
if firstword='Module' then

```


MOVING?



SUBSCRIPTION PROBLEMS?

Please don't forget to let us know. If you are having a problem with your subscription or if you are planning to move, please write to:

Amazing Computing Subscription Questions
PiM Publications, Inc.
P.O. Box 869
Fall River, MA 02722

Please remember, we cannot mail your magazine to you if we do not know where you are.

Please allow four to six weeks for processing.

```
/* Not an error - it's LC2 module size message */
'Text' instring|nl
else do
/* We assume it's some kind of error */
err_count = err_count+1
if firstword='CXERR:' then
/* This is an internal error message */
'Text Internal <<< Error number:',
line_num '>>>' * '|nl
else
/* It must be an "Operational" message */
'Text Operation <<<' instring '>>>' * '|nl
end
end
return
/* end of function ProcessOtherLine */

/* end of Proc_LC1_Out */
```

Listing Four

```
*****
*
* NextErr - finds the next error message in the
*           compiler output window and then positions
*           the cursor on the error in the source window
*
* Called by CygCC, the CygnusEd compiler invoker.
* Also can be called directly from CygnusEd editor.
* Inputs and outputs: none
*
/* Allow host to return results */
options results

/* Find the window */
if ~FindWind( 'Compiler_Window' ) then
return
else do
```

```
/* Unhighlight previous error message */
'Mark block'

/* Look for the next error message */
'Search for <<<'
if result==RESULT then do
'Okay2 No more errors... Delete Compiler_Window?'
if result then
'Quit 1'
else do
'Beg of file'
'Mark block'
end
return
end

/* Highlight the error message */
'Mark block'
'Search for >>>'; 'Right'; 'Right'; 'Right';

/* Get the contents of the error message */
'Status 55' /* contents of current line */
err_line = result

/* Parse err_line to get filename, line, and column */
parse var err_line . '>>>' fname line_num column

/* If error line contains a valid filename */
if fname!='' then do

/* Strip path from filename to get window name */
wname = right(fname,length(fname)-lastpos(':',fname))
wname = right(wname,length(wname)-lastpos('/',wname))

/* Try to find the program window */
found = FindWind( wname )

/* If there was no window corresponding to file */
if( ~found ) then do

/* Look for the source file in default dir */
if exists( fname ) then
found = 1
else do
/* Maybe it's an INCLUDE file */
if exists( 'INCLUDE:'||fname ) then do
found = 1
fname = 'INCLUDE:'||fname
end
end
/* If we were able to find the source file */
if found then do
/* Open a new window on it */
'Open new'
'Open' fname
end
else do
/* Inform the user of our troubles */
'Okay! Unable to find source file'
return
end

end

/* If we were able to find the source file */
if found then do
/* Compensate for lines added or deleted to
source file since it was compiled. We presume
here that these changes occur BEFORE the current
error location */
'Status 17' /* current lines */
lines_now = result
lines_orig = getclip( 'CygCC_NLines' )
line_num = line_num + ( lines_now - lines_orig )
/* Move cursor to location of the error */
'JumpTo' line_num column

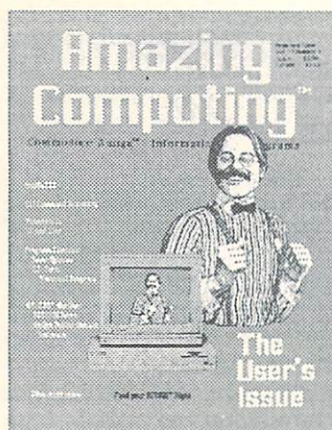
end /* end if not internal error */

end

return

/* End program NextErr */
```

•AC•



AC's BACK ISSUE INDEX



Vol. 1 No. 1, Premiere, 1986

Highlights include:

- "Super Spheres", An ABASIC Graphics Program, by Kelly Kauffman
- "Date Virus", by J. Foust
- "EZ-Term", An ABASIC terminal program, by Kelly Kauffman
- "Miga Mania", Programming fixes & mouse care, by P. Kivolowitz
- "Inside CLI", A guided insight into AmigaDos, by G. Musser

Vol. 1 No. 2, 1986

Highlights include:

- "Inside CLI: Part Two", Investigating CLI & ED, by G. Musser
- "Online and the CTS Fabite 2424 ADH Modem", by J. Foust
- "Superterm V 1.0", A terminal program in Amiga Basic, by K. Kauffman
- "A Workbench 'More' Program", by Rick Wirth

Vol. 1 No. 3, 1986

Highlights include:

- "Forth!", A tutorial
- "Deluxe Draw!!!", An AmigaBASIC art program, by R. Wirth
- "AmigaBASIC", A beginner's tutorial
- "Inside CLI: Part 3", by George Musser

Vol. 1 No. 4, 1986

Highlights include:

- "Build Your Own 5 1/4" Drive Connector", by E. Viveiros
- "AmigaBASIC Tips", by Rich Wirth
- "Scrimper: Part One", A program to print Amiga screen, by P. Kivolowitz

Vol. 1 No. 5, 1986

Highlights include:

- "The HSI to RGB Conversion Tool", Color manipulation in BASIC, by S. Pietrowicz
- "Scrimper: Part Two" by Perry Kivolowitz
- "Building Tools", by Daniel Kary

Vol. 1 No. 6, 1986

Highlights include:

- "Mailing List", A basic mail list program, by Kelly Kauffman
- "Pointer Image Editor", by Stephen Pietrowicz
- "Scrimper: Part Three", by Perry Kivolowitz
- "Optimize Your AmigaBASIC Programs For Speed", by Steve Pietrowicz

Vol. 1 No. 7, 1986

Highlights include:

- "Try 3-D", An introduction to 3-D graphics, by Jim Meadows
- "Window Requesters in Amiga Basic", by Steve Michel
- "I C What I Think", A few C graphic progs, by R. Peterson
- "Your Menu Sir!", Programming AmigaBASIC menus, by B. Catley
- "Linking C Programs with Assembler Routines", by Gerald Hull

Vol. 1 No. 8, 1986

Highlights include:

- "Computers in the Classroom", by Robert Frizelle
- "Using Your Printer With The Amiga"
- "Using Fonts from AmigaBASIC", by Tim Jones
- "Screen SaVer", Monitor protection program in C, by P. Kivolowitz
- "A Tale of Three EMACS", by Steve Poling
- "bmap File Reader in AmigaBASIC", by T. Jones

Vol. 1 No. 9, 1986

Highlights include:

- "The Loan Information Program", A BASIC program for your financial options, by Brian Catley
- "Starting Your Own Amiga-Related Business", by W. Simpson
- "Keep Track of Your Business Usage for Taxes", by J. Kummer
- "Using Fonts from AmigaBASIC: Part Two", by Tim Jones
- "68000 Macros On The Amiga", by G. Hull

Vol. 2 No. 1, January 1987

Highlights include:

- "What Digi-View Is... Or, What Genlock Should Be!", by J. Foust
- "AmigaBASIC Titles", by Bryan Catley
- "A Public Domain Modula-2 System", by Warren Block
- "One Drive Compile", by Douglas Lovell
- "A Megabyte Without Megabucks", An internal megabyte upgrade, by Chris Irving

Vol. 2 No. 2, February 1987

Highlights include:

- "The Modem", Efforts of a BBS sysop, by Joseph L. Rothman
- "The ACO Project...Graphic Teleconferencing on the Amiga", by S. R. Pietrowicz
- "Flight Simulator II: A Cross Country Tutorial", by John Rafferty
- "A Disk Librarian In AmigaBASIC", by John Kennan
- "Creating And Using Amiga Workbench Icons", by C. Hansel
- "Build Your Own MIDI Interface", by Richard Rae
- "AmigaDOS Operating System Calls and Disk File Management", by D. Haynie
- "Working with the Workbench", by Louis A. Mamakos

Vol. 2 No. 3, March 1987

Highlights include:

- "An Analysis Of The New Amiga PCs (A2000 & A500)", by J. Foust
- "Subscripts and Superscripts in AmigaBASIC", by Ivan C. Smith
- "AmigaTrix", Amiga shortcuts, by W. Block
- "Intuition Gadgets", by Harriet Maybeck Tolly
- "Forth!", Put sound in your Forth programs, by Jon Bryan
- "Assembly Language on the Amiga", by Chris Martin
- "AmigaNotes", No stereo? Y not?, by Rick Rae

Vol. 2 No. 4, April 1987

Highlights include:

- "Jim Sachs Interview", by S. Hull
- "The Mouse That Got Restored", by Jerry Hull and Bob Rhode
- "Household Inventory System in AmigaBASIC", by B. Catley
- "Secrets of Screen Dumps", by Natkun Okun
- "Amigatrix II", More Amiga shortcuts, by Warren Block

Vol. 2 No. 5, May 1987

Highlights include:

- "Writing a SoundScape Module", Programming with MIDI, Amiga and SoundScape in C, by T. Fay
- "Programming in 68000 Assembly Language", by C. Martin
- "Using FutureSound with AmigaBASIC", Programming utility with real digitized STEREO, by J. Meadows
- "Waveform Workshop In AmigaBASIC", by J. Shields
- "Intuition Gadgets: Part II", by H. MaybeckTolly

Vol. 2 No. 6, June 1987

Highlights include:

- "Modula-2 AmigaDOS Utilities", by S. Faiwizewski
- "Amiga Expansion Peripherals", by J. Foust
- "What You Should Know Before Choosing an Amiga 1000 Expansion Device", by S. Grant
- "68000 Assembly Language Programming", by Chris Martin

Vol. 2 No. 7, July 1987

Highlights include:

- "Video and Your Amiga", by Oran Sands III
- "Amigas & Weather Forecasting", by Brenden Larson
- "Quality Video from a Quality Computer", by O. Sands
- "Is IFF Really a Standard?", by John Foust
- "All About Printer Drivers", by Richard Bielak
- "68000 Assembly Language", by Chris Martin

Vol. 2 No. 8, August 1987

Highlights include:

- "Amiga Entertainment Products"
- "Modula-2 Programming"
- "Assembly Language"
- "Disk-2-Disk", by Matthew Leeds
- "Skinny C Programs", by Robert Riemersma, Jr.

Vol. 2 No. 9, September 1987

Highlights include:

- "Modula-2 Programming", Raw console dev. events, by S. Faiwizewski
- "AmigaBASIC Patterns", by Brian Catley
- "Programming with Soundscape", by T. Fay
- "Bill Volk, Vice-President Aegis Development", interview by Steve Hull
- "Jim Goodnow, Developer of Manx 'C'", interview by Harriet M Tolly

Vol. 2 No. 10, October 1987

Highlights include:

- "Max Headroom and the Amiga", by John Foust
- "Taking the Perfect Screen Shot", by Keith Conforti
- "Amiga Artist: Brian Williams", by John Foust
- "All About On-line Conferencing", by Richard Rae
- "Amiga BASIC Structures", by Steve Michel
- "Quick and Dirty Bobs", by Michael Swinger
- "Fast File I/O with Modula-2", by Steve Faiwizewski
- "Window I/O", by Read Predmore

Vol. 2 No. 11, November 1987

Highlights include:

- "Jez San Interview", StarGlider author speaks!, by Ed Bercovitz
- "Do-it-yourself Improvements To The Amiga Genlock"
- "AmigaNotes", Electronic music books, by R. Rae
- "Modula-2 Programming", Devices, I/O, & serial port, by S. Faiwizewski
- "68000 Assembly Language", by Chris Martin
- "The AMICUS Network", by John Foust
- "C Animation: Part II", by Mike Swinger
- "SoundScape Part III", VU Meter and more, by Todor Fay
- "Fun with Amiga Numbers", by Alan Barnett
- "File Browser", by Bryan Catley

Vol. 2 No. 12, December 1987

Highlights include:

- "The Ultimate Video Accessory", by Larry White
- "The Sony Connection", by Stewart Cobb

"CLI Arguments in C", by Paul Castonguay
 "MIDI Interface Adaptor", by Barry Massoni
 "Modula-2", Command line calculator, by S. Faiwizewski
 "AmigaNotes", Audio changes made in the A500 & A2000, by Rick Rae
 "Animation for C Rookies: Part III", by M. Swinger
 "The Big Picture", Assembly language programming, by Warren Ring
 "Insider/Kwikstart Review", RAM & ROM expansion: Comments & installation tips, by Ernest P. Viveiros, Sr.
 "Forth!", DumpRPort utility for your Multi-Forth toolbox, by Jon Bryan

Vol. 3 No. 1, January 1988
 Highlights include:
 "AmigaNotes", Amiga digital music generation, by Richard Rae
 "C Animation: Part IV", by Michael Swinger
 "Forth", Sorting out Amiga CHIP and FAST memory, by John Bryan
 "The Big Picture", CLI system calls and manipulating disk files, by Warren Ring
 "68000 Assembly Language Programming", Create a multi-color screen without using intuition routines, by Chris Martin
 "Modula-2 Programming", by S. Faiwizewski
 "The Ultimate Video Accessory: Part II", by L. White
 "FormatMaster: Professional Disk Formatting Engine", by C. Mann
 "BSpread", Full featured AmigaBASIC spreadsheet, by Brian Catley

Vol. 3 No. 2, February 1988
 Highlights include:
 "Laser Light Shows with the Amiga", by Patrick Murphy
 "The Ultimate Video Accessory: Part III", by L. White
 "Hooked On The Amiga With Fred Fish", by Ed Bercovitz
 "Photo Quality Reproduction with the Amiga and Digiview", by Stephen Lebars
 "Balancing Your Checkbook With WordPerfect Macros", by S. Hull
 "Solutions To Linear Algebra Through Matrix Computations", by Robert Ellis
 "Modula-2 Programming", Catching up with Calc, by Steve Faiwizewski
 "68000 Assembler Language Programming", by Chris Martin
 "AIRT", Icon-based program language, by S. Faiwizewski

Vol. 3 No. 3, March 1988
 Highlights include:
 "Desktop Video: Part IV", by Larry White
 "The Hidden Power of CLI Batch File Processing", by J. Rothman
 "A Conference With Eric Graham", edited by John Foust
 "Perry Kivolowitz Interviewed", by Ed Bercovitz
 "Jean 'Moebius' Giraud Interviewed", by Edward L. Fadigan
 "PAL Help", A1000 expansion reliability, by Perry Kivolowitz
 "Boolean Function Minimization", by Steven M. Hart
 "Amiga Serial Port and MIDI Compatibility for Your A1000", by L. Ritter and G. Rentz
 "Electric Network Solutions the Matrix Way", by Robert Ellis
 "Modula-2 Programming", The gameport device and simple sprites in action, by Steve Faiwizewski
 "The Big Picture", Unified Field Theory by Warren Ring

Vol. 3 No. 4, April 1988
 Highlights include:
 "Writing A SoundScape Patch Librarian", by T. Fay
 "Upgrade Your A1000 to A500/2000 Audio Power", by H. Bassen
 "Gels in Multi-Forth", by John Bushakra
 "Macrobatics", Easing the trauma of Assembly language programming, by Patrick J. Horgan
 "The Ultimate Video Accessory: Part V", by Larry White
 "The Big Picture, Part II: Unified Field Theory", by Warren Ring

Vol. 3 No. 5, May 1988
 Highlights include:
 "Interactive Startup Sequence", by Udo Pernisz
 "AmigaTrix III", by Warren Block
 "Proletariat Programming", Public domain compilers, by P. Quaid
 "The Companion", Amiga's event-handling capability, by P. Gosselin

"The Big Picture, Unified Field Theory: Part III", by Warren Ring
 "Modula-2", Termination modules for Benchmark and TDI compilers, by Steve Faiwizewski
 "68000 Assembly Language", Peeling away the complication of display routines, by Chris Martin
 "The Command Line: The First Installment", by Rich Falconburg

Vol. 3 No. 6, June 1988
 Highlights include:
 "Reassigning Workbench Disks", by John Kennan
 "An IFF Reader in Multi-Forth", by Warren Block
 "Basic Directory Service Program", Programming alternative to the GimmeZeroZero, by Bryan Catley
 "C Notes from the C Group", A beginner's guide to the power of C programming, by Stephen Kemp
 "An Amiga Forum Conference with Jim Mackraz", The Amiga market as seen by the "Stepfather of Intuition."
 "The Command Line: Exploring the multi-talented LIST command", by Rich Falconburg

Vol. 3 No. 7, July 1988
 Highlights include:
 "An Interview with 'Anim Man', Gary Bonham" by B. Larson
 "Roll Those Presses!", The dandy, demanding world of desktop publishing, by Barney Schwartz
 "Linked Lists in C", by W. E. Gammill
 "C Notes from the C Group", The unknown "C" of basic object and data types, by Stephen Kemp

Vol. 3 No. 8, August 1988
 Highlights include:
 "The Developing Amiga", A gaggle of great programming tools, by Stephen R. Pietrowicz
 "Modula-2 Programming", Libraries and the FFP and IEE math routines, by Steve Faiwizewski
 "C Notes from the C Group: Arrays and pointers unmasked", by Stephen Kemp
 "TrackMouse", Converting a standard Atari trackball into a peppy Amiga TrackMouse, by Darryl Joyce
 "Amiga Interface for Blind Users", by Carl W. Mann
 "Tumblin' Tots", Assembly language program, by David Ashley
 Plus—A Look At Amiga Entertainment

Vol. 3 No. 9, September 1988
 Highlights include:
 "The Kideo Tapes", A Georgia elementary school puts desktop video to work, by John Dandurand
 "Speeding Up Your System", Floppy disk caching, by Tony Preston
 "Computer-Aided Instruction", Authoring system in AmigaBASIC, by Paul Castonguay
 "Gels in Multi-Forth, Part II: Screenplay", by John Bushakra
 "AmigaNotes: How IFF sound samples are stored", by Richard Rae
 "C Notes from the C Group", Operators, expressions, and statements in C uncovered, by Stephen Kemp

Vol. 3 No. 10, October 1988
 Highlights include:
 "The Command Line: NEWCLI: A painless way to create a new console window", by Rich Falconburg
 "Record Keeping for Freelancers: A Superbase Professional Tutorial", by Marion Deland
 "On The Crafting of Programs", Optimization kicks off our series on programming savvy, by David J. Hankins
 "Bob and Ray Meet Frankenstein", Create, animate, and metamorphose graphics objects in AmigaBASIC, by Robert D'Asto
 "Digital Signal Processing in AmigaBASIC", Perform your own digital experiments with Fast Fourier Transforms, by Robert Ellis
 "HAM & AmigaBASIC", Pack your AmigaBASIC progs with many of the Amiga's 4096 shades, by Bryan Catley
 "CAI—Computer Aided Instruction: Part II", by Paul Castonguay

Vol. 3 No. 11, November 1988
 Highlights include:
 "Structures in C", by Paul Castonguay
 "On The Crafting of Programs", Speed up your progs, by D. Hankins
 "Desktop Video VI: Adding the Third Dimension", by L. White

"More Linked Lists in C: Techniques and Applications", Procedures for managing lists, storing diverse data types in the same list, and putting lists to work in your programs, by Forest W. Arnold
 "BASIC Linker", Combine individual routines from your program library to create an executable program, by Brian Zupke

Vol. 3 No. 12, December 1988
 Highlights include:
 "The Command Line: What to do when the commands of AmigaDos fail", by Rich Falconburg
 "Converting Patch Librarian Files", by Phil Saunders
 "The Creation of Don Bluth's Dragon's Lair", by Randy Linden
 "Easy Menus in JForth", by Phil Burk
 "Extending AmigaBASIC", The use of library calls from within AmigaBASIC, by John Kennan
 "Getting Started In Assembly", by Jeff Glatt
 "C Notes From The C Group: Program or function control coding", by Stephen Kemp
 "AmigaDos, Assembly Language, And FileNotes", Weapons in the war against file overload; accurate, descriptive file naming, by Dan Huth

Vol. 4 No. 1, January 1989
 Highlights include:
 "Desktop Video", by Richard Starr
 "Industrial Strength Menus", by Robert D'Asto
 "Scrolling Through SuperBitMap Windows", by Read Predmore
 "Sync Tips: Dot crawl, the Amiga and composite video devices", by Oran J. Sands
 "Stop-Motion Animation On The Amiga", by Brian Zupke
 "The Command Line: New and Improved Assembly Language Commands", by Rich Falconburg
 "Pointers, Function Pointers, and Pointer Declarations in C", by Forest W. Arnold
 "Death of a Process", Developing an error-handling module in Modula-2, by Mark Cashman

Vol. 4 No. 2, February 1989
 Highlights include:
 "Max Morehead Interview", by Richard Rae
 "A Common User Interface for the Amiga", by Jim Bayless
 "SPY: Programming Intrigue In Modula -2", by Steve Faiwizewski
 "Sync Tips: Getting inside the genlock", by Oran Sands
 "On the Crafting of Programs: A common standard for C programming?", by D. J. Hankins
 "C Notes from the C Group: An introduction to unions", by Steven Kemp
 "The Command Line: Your Workbench Screen Editor", by Rich Falconburg
 "An Introduction to ARexx programming", by Steve Faiwizewski

Vol. 4 No. 3, March 1989
 Highlights include:
 "Fractal Fundamentals", by Paul Castonguay
 "Image Processing With Photosynthesis", by Gerald Hull
 "Benchmark 1: Fully Utilizing The MC68881", Part I: Turbocharging the savage benchmark, by Read Predmore
 "Breaking the Bmap Barrier", Streamline AmigaBASIC library access with Quick—Lib, by Robert D'Asto
 "Double Play", AmigaBASIC program yields double vision, by Robert D'Asto
 "The Video Desk: The Amiga meets Nikon Camera", by Larry White

Vol. 4 No. 4, April 1989
 Highlights include:
 "AmiEXPO Art and Video Contest Winners", by Steve Jacobs
 "Adding the Not-So-Hard Disk", by J. P. Twardy
 "The Max Hard Drive Kit", A hard drive installation project, using Palomax's Max kit, by Donald W. Morgan
 "Sync Tips: A clearer picture of video and computer resolutions", by Oran J. Sands
 "Passing Arguments", Step-by-step on how to pass data from the CLI to AmigaBASIC, by Brian Zupke
 "Creating a Shared Library", by John Baez

Vol. 4 No. 5, May 1989
 Highlights include:
 "The Business of Video", by Steve Gillmor

"An Amiga Adventure", The globetrotting Amiga in Cologne, Germany, by Larry White
 "Uninterruptible Power Supply (UPS), Part I", by Steve Bender
 "Building Your Own Stereo Digitizer", by Andre Theberge
 "MIDI Out Interface", by Br. Seraphim Winslow
 "Digitized Sounds in Modula-2", by Len A. White
 "Sync Tips: The secrets hidden beneath the flicker mode", by Oran J. Sands
 "Insta Sound in AmigaBASIC", by Greg Stringfellow
 "C Notes from the C Group: Formatted output functions", by Stephen Kemp

Vol. 4 No. 6, June 1989
 Highlights include:
 "Adventures in AREXX", by Steve Gillmor
 "At Your Request: Design your own requesters in AmigaBASIC", by John F. Weiderhorn
 "Exploring Amiga Disk Structures", by David Martin
 "Diskless Compile in C", by Chuck Raudonis
 "(UPS), Part II", by Steve Bender
 "Programming the '881 Part II", A discussion on how to calculate Mandelbrot & Julia sets, by Read Predmore
 "C Notes from the C Group: Ways to avoid problems when passing parameters between functions", by Stephen Kemp

Vol. 4 No. 7, July 1989
 Highlights include:
 "An Inside look at UltraCard", by Steve Gillmor
 "Adapting Analog Joysticks to the Amiga", by David Kinzer
 "Using Coordinate Systems: Part II of the Fractals series addresses the basis of computer graphics", by Paul Castonguay
 Plus—A Look At Amiga Entertainment

Vol. 4 No. 8, August 1989
 Highlights include:
 "Getting Started in Video", by Richard Starr
 "C Notes: Directing programs via the Command Line", by Stephen Kemp
 "Executing Batch Files in AmigaBASIC", by Mark Aydelotte
 "Building a Better String Gadget", by John Bushakra
 "On Your Alert: Using System Alerts from BASIC", by John F. Wiederhorn

Vol. 4 No. 9, September 1989
 Highlights include:
 "Digitizing Color Slides And Negatives on the Amiga", by Ron Gull
 "Improving Your Graphics Programming", by Richard Martin
 "Cell Animation in Modula-2", by Nicholas Cirasella
 "More Requesters in AmigaBASIC", by John R. Wiederhorn
 "DeluxePaint III — The Inside Story", EA's Dan Silva tells how DeluxePaint III evolved, by Ben & Jean Means
 "Amiga In Desktop Presentation", Presentation techniques to enhance your meetings and seminars, by John Steiner
 "Multitasking in Fortran", by Jim Locker
 "Gels In Multi-Forth: Part III", by John Bushakra

Vol. 4 No. 10, October 1989
 Highlights include:
 "BetterTrackMouse", A true one-handed trackball mouse, by Robert Katz
 "Conference with Will Wright and Brian Conrad of SimCity fame", edited by Richard Rae
 "A1000 Rejuvenator, Conference with Gregory Tibbs", edited by Richard Rae
 "APL & the Amiga", by Henry Lippert
 "Saving 16-color pictures in high-resolution", Part Three of the Fractals series, by Paul Castonguay
 "More requesters in AmigaBASIC", by John Wiederhorn
 "Glatt's Gadgets", Adding gadgets in Assembly, by Jeff Glatt
 "Function Evaluator in C", by Randy Finch
 "Big Machine On Campus", Humboldt State University in Northern California goes Amiga, by Joel Hagen
 "Typing Tutor", by Mike "Chip" Morrison

Vol. 4 No. 11, November 1989
 Highlights include:
 "The Amiga Hardware Interface", by John Iovine
 "The Command Line: Examine the features in the AmigaDOS 1.3 Enhancer software package", by Rich Falconburg
 "C Notes from the C Group: Creating your own libraries

in C", by Stephen Kemp
 "APL & the Amiga, Part II", by Henry Lippert
 "FastPix()", A faster pixel-drawing routine for the Aztec C compiler, by Scott Steinman
 "64 Colors in AmigaBASIC", by Bryan Catley
 "Fast Fractals", Generate Mandelbrot Fractals at lightning speed, by Hugo M.H. Lyppens
 "Multitasking in Fortran", by Jim Locker

Vol. 4 No. 12, December 1989
 Highlights include:
 "The MIDI Must Go Thru", by Br. Seraphim Winslow
 "View From the Inside: Bars&Pipes", Bars&Pipes designer gives a tour of Blue Ribbon Bakery's music program, by Melissa Jordan Grey
 "AREXX Part II", by Steve Gillmor
 "A CLI Beginner's Questions Answered", by Mike Morrison
 "Trees and Recursion", by Forest W. Arnold
 "C Notes from the C Group", A look at two compressing data techniques, by Stephen Kemp
 "The Command Line: Exploring commands in AmigaDOS", by Rich Falconburg
 "Amiga Circuits", The techniques required to input information via the parallel port, by John Iovine

Vol. 5 No. 1, January 1990
 Highlights include:
 "The Making Of The 1989 BADGE Killer Demo Contest Winner, The Sentinel", by Bradley W. Schenck
 "Animation For Everyone", by Barry Solomon
 "Animation With Sculpt-Animate 4D", by Lonnie Watson
 "Animation? BASICally!", Using Cell animation in AmigaBASIC, by Mike Morrison
 "Menu Builder", Building menus with Intuition, by Tony Preston
 "Facing the CLI", Disk structures and startup-sequences, by Mike Morrison
 "Dual Demo", Programming an arcade game, by Thomas Eshelman
 "Scanning The Screen", Part Four in the Fractals series, by Paul Castonguay
 "It's Colder Than You Think", Calculating the wind chill temperature, by Robert Klimaszewski

Vol. 5 No. 2, February 1990
 Highlights include:
 "A Beginner's Guide to Desktop Publishing On The Amiga", by John Steiner
 "A Desktop Publishing Primer", Clearing up some of the mystery surrounding printers.
 "Resizing the shell/CLI Window", by William A. Jones
 "Call Assembly Language from BASIC", by Martin F. Combs
 "You Too Can Have A Dynamic Memory", Flexible string gadget requester using dynamic memory allocation, by Randy Finch
 "An Amiga Conundrum", An AmigaBASIC program for a puzzle-like game, by David Senger
 "View From The Inside: ScanLab", ASDG's President shares the development of ScanLab, by Perry Kivolowitz
 "AMIGANET", by Ernest P. Viveiros, Jr.

Vol. 5 No. 3, March 1990
 Highlights include:
 "Screen Aid", A quick remedy to prolong the life of your monitor, by Bryan Catley
 "An Introduction to MIDI", by R. Shams Mortier
 "The Other Guys' Synthia Professional", review by David Duberman
 "Passport's Master Tracks Pro vs. Blue Ribbon Bakery's Bars&Pipes", by Ben Means
 "Microillusions' Music-X", review by Rob Bryantton
 "Diemer Development's C-ZAR", review by R. Shams Mortier
 "Dr. T's Keyboard Controlled Sequencer", review by Phil Saunders
 "MusicTidler", Generating a titler display to accompany the audio on a VCR recording, by Brian Zupke

Vol. 5 No. 4, April 1990
 Highlights include:
 "Handling MS-DOS Files", Adapting your Amiga to MS-DOS using a 5.25" disk drive, by Jim Locker
 "Bridging the 3.5" Chasm", Making Amiga 3.5" drives compatible with IBM 3.5" drives, by Karl D. Belsom
 "Bridgeboard Q & A", by Marion Deland

"Handling Gadget & Mouse IntuiEvents", More gadgets in Assembly, by Jeff Glatt
 "Ham Bones", Programming in HAM mode in AmigaBASIC, by Robert D'Asto
 "Gambling with your video, Amiga-style", Problems with trading genlocks with your friends, by Oran Sands
 "Distant Suns", review by Mike Hubbard

Vol. 5 No. 5 May 1990
 Highlights include:
 "Commodore's Amiga 3000", preview
 "Newtek's Video Toaster", preview
 "Getting started With Deluxe Video III", tutorial by David Johnson
 "Do It By Remote", Building an Amiga-operated remote controller for your home, by Andre Theberge
 "Turn Your Amiga 1000 Into A ROM-based Machine", by George Gibeau Jr. & Dwight Blubaugh
 "Super Bitmaps In BASIC", Holding a graphics display larger than the monitor screen, by Jason Cahill
 "Rounding Off Your Numbers", by Sedgewick Simons Jr.
 "Faster BASIC Mouse Input", by Michael S. Fahrion
 "Print Utility", by Brian Zupke

Vol. 5 No. 6, June 1990
 Highlights include:
 "Convergence", Part Five of the Fractal series, by Paul Castonguay
 "C++: An introduction to object-oriented Amiga programming", by Scott B. Steinman
 "APL and the Amiga: Primitive Functions and Their Execution", by Henry T. Lippert
 "Amiga Turtle Graphics", by Dylan McNamee
 "Building A Rapid Fire Joystick", by John Iovine
 "The AM 512", Upgrade your A500 to a 1 megabyte machine, by James Bentley
 "PageStream 1.8", review by John Steiner
 "WordPerfect Macros", by Mike Hubbard
 "Mail Order Macros", Addressing envelopes using WordPerfect macros, by Armando Cardenas
 "DigiMate III", review by Frank Mc Mahon

Vol. 5 No. 7, July 1990
 Highlights include:
 "Commodore announces CDTV"
 "Apples, Oranges, and MIPS: 68030-based Accelerators For The Amiga 2000", by Ernest P. Viveiros, Jr.
 "Pixound", review by R. Shams Mortier
 "Hyperchord", by Howard Bassen
 "Exceptional Conduct", Quick response to user requests, through efficient program logic, by Mark Cashman
 "Poor Man's Spreadsheet", A simple spreadsheet program that demonstrates manipulating arrays, by Gerry L. Penrose
 "Tree Traversal and Tree Search", Two methods for traversing trees, by Forest W. Arnold
 "Crunchy Frog II", by Jim Fiore
 "Getting to the Point: Custom Intuition Pointers In AmigaBASIC", by Robert D'Asto
 "Synchronicity: Right & Left Brain Lateralization", by John Iovine
 "Snap, Crackle, & POP!", Fixing a monitor bug on Commodore monitors, by Richard Landry

Vol. 5 No. 8, August 1990
 Highlights include:
 "Mimetics' FrameBuffer", review by Lonnie Watson
 "The VidTech Scanlock", review by Oran Sands
 "Amigas in Television", The Amiga in a cable television operation, by Frank McMahon
 "Desktop Video in a University Setting", The Amiga at work at North Dakota State University, by John Steiner
 "Credit Text Scroller", review by Frank McMahon
 "Graphic Suggestions", Other ways to use your Amiga in video production, by Bill Burkett
 "Title Screens That Shine: Adding light sources with DeluxePaint III", by Frank McMahon
 "The Amiga goes to the Andys", by Curt Kass
 "Breaking the RAM Barrier", Longer, faster, smoother animations with only one meg of RAM, by Frank McMahon
 "Fully Utilizing the 68881 Math Coprocessor: Timings and Turbo_Pixel functions", by Read Predmore
 "APL and the Amiga: Part IV", by Henry T. Lippert
 "Sound Quest's MidiQuest", review by Hal Belden

The Fred Fish Collection

Due to the increasing size of the Fred Fish Collection, only the latest disks are represented here. For a complete list of all AC, AMICUS, and Fred Fish Disks, cataloged and cross-referenced for your convenience, please consult the current AC's Guide To The Commodore Amiga available at your local Amazing Dealer.

Fred Fish Disk 335

BoingDemo Demo version of a neat game due for release in March 1990. It is fully functional but the play time is limited to five minutes per play. Version 0.30, binary only. Author: Kevin Kelm, Alternate Realities

DTC A utility providing a simple calendar which can hold and show appointments. It may be useful in managing your time. Its chief goals were to provide day, week and month at a glance for any date between 1/1/0001 and 12/31/9999, defaulting to the current date. It is menu driven and fairly easy to use. Includes source in Fortran. Author: Mitch Wyle, Amiga port by Glenn Ewert

SeeHear A program to do a spectrogram of a sampled sound file. This is a graph with time on one axis, frequency on the other and the sound intensity at each point determining the pixel color. With source in C, including FFT routine. This is version 1.1. Author: Daniel T. Johnson

Fred Fish Disk 336

Car A two-dimensional full screen scrolling racing game with realistic four channel stereo sound and overscan, for either NTSC or PAL Amigas. The goal is to guide your car around one of ten selected tracks. Each track has its individual high score list. Version 2.0, binary only. Author: Anders Björn

FileWindow A completely public domain file requester which may be used in any program, even commercial ones. It uses dynamically allocated memory to hold the file names so the only limitation is the amount of memory available. Includes a filter option to limit display of filenames to only ones with a specific extension. Names are automatically sorted while they are being read and displayed. V1.10, includes source. By: Anders Björn

MiniBlast A shoot'em up game which runs just fine in a multi-tasking environment. At last you can enjoy a satisfying megablast while you are writing a boring essay. Shoot anything that moves, and if it doesn't move, shoot it anyway. V1.00, binary only. By: Anders Björn

Sys A game built on the addictive game PONGO but with several added features. You have been assigned the demanding task of clearing viruses from your SYSDOP's hard disk. To kill a virus, you simply kick a disk at it. There are fifty different levels, and on each level, the speed will increase and the viruses will be smarter and start to hunt you. V2.10, binary only. By: Anders Björn

Fred Fish Disk 337

CManual A complete C manual for the Amiga which describes how to open and work with screens, windows, graphics, gadgets, requesters, alerts, menus, IDCMP, sprites, etc. The manual consists of more than 200 pages in 11 chapters, together with more than 70 fully executable examples with source code. When unpacked, the manual and examples nearly fill up three standard Amiga floppies. This is version 1.00 and includes source for all examples. Author: Anders Björn

Fred Fish Disk 338

Cop This is a copy of the Decus cpp, ported to the Amiga. This is a more powerful and complete than either of the built in cpp's in Manx or Lattice C. This is an update to the version on disk 28. It has had some ANSI features added. Includes source. By: Martin Minow, Olaf Seibert

SASTools Various submissions from "Sick Amiga Soft". Includes some virus tools, some screen hacks, some small games, and miscellaneous utilities. Includes source in assembly and Modula-II. Author: Jörg Sixt

SID A very comprehensive directory utility for the Amiga that supports at least a couple of dozen different commands for operating on files. Version 1.06, binary only. Author: Timm Martin

Fred Fish Disk 339

PCQ A freely redistributable, self compiling, Pascal compiler for the Amiga. The only major feature of Pascal that is not implemented is sets. This is version 1.1c, an update to version 1.0 on disk 183. It is much enhanced and about four times faster. Includes the compiler source and example programs. Author: Patrick Quaid

Fred Fish Disk 340

NorthC A complete freely redistributable C environment for the Amiga based on the Soccobon Ltd C compiler, Charlie Gibb's assembler, the Software Distillery's

Pilot linker, and portions from other sources. Steve has pulled everything together and added some enhancements in the process. Version 1.0, partial source only. By: Steve Hawtin, et al.

SpeakerSim Demo version of SpeakerSim 2.0, a loudspeaker CAD program. Simulates vented (Thiele-Small) and closed box systems. Also simulates 1st, 2nd, and 3rd order high and low pass filters. Binary only. By: Dissidents

SpeakerSim Demo version of SpeakerSim 2.0, a loudspeaker CAD program. Simulates vented (Thiele-Small) and closed box systems. Also simulates 1st, 2nd, and 3rd order high and low pass filters. Binary only. By: Dissidents

Fred Fish Disk 341

P2C P2C is a tool for translating Pascal programs into C. It handles the following Pascal dialects: HP Pascal, TurboUCSD Pascal, DEC VAX Pascal, Oregon Software Pascal/2, Macintosh Programmer's Workshop Pascal, Sun/Berkeley Pascal. Modula-2 syntax is also supported. Most reasonable Pascal programs are converted into fully functional C which will compile and run with no further modifications. V1.13 Includes source. Author: Dave Gillespie, Amiga port by G. R. (Fred) Walter

Fred Fish Disk 342

IE This is an icon editor which can create and modify icons up to 640x200 pixels in size (also dual render). It can set stack size, position of icon (also free-floating), default tool, 10 tool types and control over opened window. It can also generate the C source code behind the icon for program inclusion. Version 1.0, binary only, source available from author. Author: Peter Kiem

SKsh A ksh-like shell for the Amiga. Some of its features include command substitution, shell functions with parameters, aliases, local variables, local functions, local aliases, powerful control structures and tests, emacs style line editing and history functions, I/O redirection, pipes, large variety of built-in commands, Unix style wildcards, Unix style filename conventions, filename completion, and coexistence with scripts from other shells. Very well documented. Version 1.4, an update to version 1.3 on disk 309. New features include a "tiny" version, a working case construct, support for resident commands, smaller and faster external commands, and more. Binary only. Author: Steve Koren

Softfont Converts portrait soft fonts for HP LaserJet compatible laser printers to landscape format. This is an update to FF237. Includes source. Author: Thomas Lynch

Fred Fish Disk 343

SnakePit A simple, yet addictive game in which you must get the snake (you) off of the screen. There are, however, some rough spots and some obstacles that may need to be overcome. Excellent example of a game that is as system friendly as possible (with source). By: Michael Sinz

SoftSpan Soft Span BBS program. Intuitive, command-line based menu system with message bases, up/down loads, file credit system, extensive help system, etc. This is shareware version 1.0, binary only, Lattice C source code available from the author. Author: Mark Wollschlaeger

StockBroker A program that helps you follow the recent table of exchange from (one or more) share(s). But of course you must tell the Amiga the recent table of exchange every day. Requires AmigaBASIC. Binary only. Author: Michael Hanelt

Fred Fish Disk 344

Keyboard Functions to translate RAWKEY Intuition messages into usable keycodes. Translation into Modula-2 of C source (by Fabian G. Duloe, III) on disk 291. Version 1.0. Includes source. Author: Fabian G. Duloe III, Peter Graham Evans

RKMCompanion

A two disk set of material created by Commodore for use with the 1.3 revision of the Amiga ROM Kernel Reference Manual, Libraries and Devices, published by Addison-Wesley. Almost 300 files, including C source code examples and executables, have been packed into two hard archives, one for each disk of the two disk set. These examples are not public domain, but may be used and distributed under the conditions specified in the copyrights. Author: Commodore Business Machines, Inc.

Fred Fish Disk 345

CRobots A game based on computer programming. Unlike arcade type games which require human input controlling some object, all strategy in CRobots is condensed into a C language program that you design and write, to control a robot whose mission is to seek out, track, and destroy other robots, each running different programs. All robots are equally equipped, and up to four may compete at once. This is version 2.3w, an update to FF331. Binary only, source available from author. Author: Tom Poindexter, Amiga version by David Wright

Du Prints number of disc blocks used in selected files or directories. Modified from original version on disk 48 to make output more readable, and handle 'C' exit. Includes source. By: Joe Mueller, enhancements by Gary Duncan

GetImage An enhanced version of "gi" from disk 14. It now looks for the GRAB marker, in the brush file, instead of assuming that it is at a specific place, sets up the PlanePick value in the image structure, and deletes any unused bitplanes to save memory and disk space. Includes source. Author: Mike Farren, enhancements by Chuck Brand

MemFrag Displays number of memory chunks/sizes to show memory fragmentation. Chunks are displayed as 2^n bytes which is a rough guide but still useful. This is an enhanced version of "Frag" from disk 69. Includes source. By: Mike Meyer, enhancements by Gary Duncan

Roses

A program that draws sine roses. Implements an algorithm given in the article "A Rose is a Rose ..." by Peter M. Maurer in American Mathematical Monthly, Vol 94, No. 7, 1987, p 631. A sine rose is a graph of the polar equation $r = \sin(n \cdot d)$ for various values of n and d. Author: Carmen Arino

Unshar

This program extracts files from Unix shar archives. It scores over similar programs by being small and fast, handling extraction of subdirectories, recognising a wide variety of 'sed' and 'cat' shar formats, and handling large files spread across several shar files. This is version 1.3, an update to the version on disk 287. Includes C source. Author: Eddy Carroll

VcEd

A Voice (Tone) Editor for the Yamaha 4 Operator series synthesizers. Binary only, source available from author. Author: Chuck Brand

X2X Cross converts between Motorola/Intel/Tektronix ASCII-hex files. These files are typically used for down-loading into EPROMs, or for transmission where binary files cause chaos. Handles S1, S2, S3, INTEL (inc USB4 records), Tektronix (inc extended). Source included. Author: Gary Duncan

Fred Fish Disk 346

Az A nice little text editor that is fast, simple to use, and very Amigaized. This is version 1.50, an update to FF 228, with lots of new features, bug fixes, and other improvements. Binary only. By: Jean-Michel Forgeas

CassEt Cassette tape label printer. Includes source in GFA Basic. Author: Thorsten Ludwig

FME Patch to AllocMem() to allow badly designed programs which request fast mem without necessity to be run on 512k machines. Includes source in assembler. Author: Holger Lubitz

GoWB Very small (296 bytes) and effective replacement for the well known "LoadWB" and "EndCL" command pair. This release fixes a severe bug in the first version which used to guru if run out of a script. Includes source in C. Author: Oliver Wagner

PacketSupport

A link library, for use with Lattice C, providing a few functions to handle DOS packet postage. Includes source. Author: Oliver Wagner

PatchNTSC OS fix to allow the growing number of PAL display programs to be run on NTSC machines. Will patch the Intuition OpenScreen() function to assure screens with PAL height to be opened in interface mode. Includes source in assembler. Author: Oliver Wagner

TextPaint

Second major release of the Ansi editor. All major bugs have been fixed, and a bunch of new options have been added, e.g. possibility to reload ansi files or CLI modules, 4 color option, optimized keyboard layout, new drawing modes, right mouse button support (like DeluxePaint) and much more. Binary only, shareware. By: Oliver Wagner

Timetest

Working example to show the time() and gmtime() functions of the Lattice C support library. Includes source in C. Author: Oliver Wagner

WBD

Possibly the smallest utility to set the workbench screen to any depth. Includes source in C. By: Oliver Wagner

Fred Fish Disk 347

Cursor A 3-pass BASIC Compiler for BASIC programs written in AmigaBASIC, does not yet support all of the BASIC commands but is able to compile itself. This is version 1.0, includes source. Author: Jürgen Förster

Drip Drip is an arcade style game with 15 floors (levels). You must move along the pipes of each floor and rust them to advance to the next level. Every 3 floors completed will entitle you to a bonus round where extra drips can be won. An extra drip will also be awarded for every 10,000 points. Binary only. Author: Art Skiles

Fred Fish Disk 348

ColorReq Describes the update to the color.library and has an example program, with source, that demonstrates its use. Author: Dissidents Software

DisEditor This is a demo of the dissidents shareware text editor. Version 1.1, binary only. Author: Dissidents Software

DisSecretary

This program can be used to file information in a "file cabinet" type environment. It is well suited

for jobs such as maintaining a disk catalog, or user group membership, etc. Included is a data file of the library catalog, disks 1 to 310. Version "Wanda", binary only. Author: Dissidents Software

FileIO Contains updated files for version 1.6 of the dissidents requester library. There is a bug fix to the library as well as a new function. See FF257 for the complete documentation, and examples. By: Dissidents Software

ILBMLib Contains updated files for the dissidents ibm.library on FF237, with new lib features and a new library. Also included is a much improved (better organized) doc file, and new C examples that show how to use the library for any kind of IFF file. See FF237 for other examples. Author: Dissidents Software

InstallLibs A program to copy files to the LIBS: dir of a boot disk. Can be used to create a handy installation program (hard disks especially) for programs that need disk-based libraries. Includes source. By: Dissidents Software

SAMP An IFF sampled sound format designed for professional music use. It can be used for 16-bit samples, multiple waveforms, etc. Includes a SAMP reader/writer shared library, interface routines, and programming examples. Also includes a program to convert 8SVX to SAMP. Author: Dissidents Software

Fred Fish Disk 349

MED A music editor much like SoundTracker. A song consists of up to 50 blocks of music, which can be played in any order. Editing features include cut/paste/copy tracks or blocks, changing the vibrato, tempo, crescendo, and note volume. Other features include switching of the low-pass-filter on or off on a per song basis, and a cute little animated pointer of a guy doing "jumping jacks" in time to the music! Version 2.00, an update to version 1.12 on FF255. Now includes full source. Author: Teijo Kinnunen

Fred Fish Disk 350

Icons A large variety of icons for many uses, of practically every description. Most are animated. By: Bradley W. Schenck

MemMoneter A program that opens a narrow window and graphically displays your memory usage like a gauge. Based on WFRags, by Tomas Rokicki. Version 2.10, includes source. Author: Howard Hull

Stitchery This shareware program loads in IFF images and creates charted patterns from them for use in counted cross-stitch and other forms of needlework. It requires one megabyte of memory to run, and works best with a good high-resolution printer for printing the patterns. The Stitchery was written with The Director and the Projector is included. Version 1.21. Author: Bradley W. Schenck

TrackUtils

Two utilities that deal with disk tracks. TCopy copies one or more tracks from one disk to another, and is useful for copying part of a floppy disk into RAD: during bootup. TFile creates a dummy file which "marks" a specified range of tracks, preventing AmigaDOS from using them and allowing them to be used for raw trackdisk data. Includes C source. Author: Eddy Carroll

Fred Fish Disk 351

PDC Publicly Distributable C (PDC) is a complete C compilation system including a compiler, assembler, linker, librarian, and numerous utilities, documentation files, libraries, and header files. PDC supports many ANSI features including all ANSI preprocessor directives, function prototyping, structure passing and assignment. In addition it supports Lattice C compatible local pragmas, precompiled header files, builtin functions, and stack checking code. V3.33 includes source. By: Lionel Hummel, Paul Petersen, et al.

Fred Fish Disk 352

MG Beta version of mg3, including ARExx support. This is probably the most stable beta for the next year, as many new features are going in after this. Amiga-only release. Sources compressed with harc to fit on the disk. Update to FF147. Author: Mike Meyer, et al.

PrintHandler

A custom PRT: driver which offers easy single sheet support as well as limited data spooling. Version 1.6, an almost entirely rewritten update to FF282. Includes source in C. Author: Olaf Barthel

TreeWalk File tree walking subroutine designed to be fast, robust, and not use a lot of any critical resource. Includes both a CLI interface to that routine the form of a find-like utility that uses C expressions instead of Unix-like flags, and a program to tell you if directory trees will fit on a given disk, or how many extra blocks you'll need if they won't. Includes source. Update to FF289. Author: Mike Meyer

Fred Fish Disk 353

AztecArp An Arp package fixed to work with the 5.0 release of the Aztec 'C' compiler. The original Manx support files were incomplete, contained bugs preventing them from working properly and had the wrong linker format. Includes source. Author: Olaf Barthel

CompDisk A disk compression/disk compression package which was written to be fast and easy to use. Includes an Arp and an Intuition interface. Includes source in C. Author: Olaf Barthel

NorthC	A complete freely redistributable C environment for the Amiga based on the Sozobon Ltd C compiler, Charlie Gibb's assembler, the Software Distillery's linker, and portions from other sources. Steve has pulled everything together and added some enhancements in the process. This is version 1.1, an update to version 1.0 on disk 340. Partial source only. Author: Steve Hawtin, et al.	OPSSc	OPSSc is a compiler for the expert system language OPS5. The compiler takes OPS5 source code as input and creates a C source code file to be compiled to create an executable. Arbitrary C code may be linked with the executable and executed as a result of finding rules. The system's strong point is its speed and as a result it sometimes has large executables and large memory requirements. At least 1 Meg. of memory is suggested. Binaries only for compiler and run-time library. Version 1.08a. Requires a C compiler. Authors: Bernie J. Lofaso, Jr., Dan Miranker and Arun Chandra.
Fred Fish Disk 354		Pipeline	A game like the commercial game "Pipe dream" (Pipe mania). Features a joystick and PAL display. High scores are saved to disk. Version 1.0, includes source. Author: Andre Wichmann.
FastBlit	A small tool to speed up blitter operations by up to 60%. Version 1.0, binary only. Author: Ralf Thanner	ReDate	Scans a disk and dates each directory according to the most recent item contained within (not including .info files). Ideal for use after a COPY ALL CLONE, where the directories are CREATED rather than copied and thus lose their date information. Includes source in assembler. Author: Jim Butterfield
KeyMacro	A keyboard macro program, configurable via a text file, that also supports hotkey program execution. You can map up to eight functions to each key, including keys such as cursor keys, the return key, etc. Version 1.4, an update to version 1.0 on disk 325, which fixes the bugs in version 1.0. Includes source in C. Author: Olaf Barthel	RoadRoute	Revision of trip planner program to find "best road route" between any two points of travel. The user is encouraged to customize files CITIES and ROADS to suit travel interests. This is version 1.5, an update to the original version on disk 251, and makes provision for very large city menus and itineraries. You might like to use files from disk 328 (Mayes/Delzer). Also includes RoadScan, a checker for RoadRoute files (CITIES and ROADS). Very large files may contain goofs (cities with no roads, the same road entered twice, etc.), or oddities (direct road not as fast as multi-point). These are pointed out, together with areas where users might wish to make economies in the data base. Includes source in C. Author: Jim Butterfield
MandelMountains	A program that renders three-dimensional images of blowups of the Mandelbrot set. Includes several example images. This is version 2.0, an update to version 1.1 on disk 295. Shareware, binary only. Author: Mathias Ortmann	ScanIFF	Scans through an IFF file, identifying the elements. Faster than standard utility IFFCheck since it uses Seek, but does not do IFFCheck's detailed format checking. Intended for use as a "template" from which programmers can code their specific application. For example, an expanded version has been used to extract instrument data from music files. Includes source in assembler. Author: Jim Butterfield
MemGuard	A MemWatch like program which has been rewritten in assembly language for maximum speed and efficiency. Unlike MemWatch, MemGuard does not run as task in a dummy loop but rather as a low-level interrupt routine which is capable of trapping memory trashing even before exec might know of it and even while task switching is forbidden. Version 1.1a, an update to version 1.0 on disk 325, binary only. Author: Ralf Thanner	ViewDir	A LIST type of utility showing contents of a disk or directory. For directories, shows SIZE. For files, takes a quick look and identifies TYPE if possible. Update to original version on disk 251. Now works with SPAT for pattern matching, and has a small style change. Includes source in assembler. Author: Jim Butterfield
MXMLib	An example Amiga shared library compiled with Aztec C 5.0. This library contains basic support functions employed by programs such as KeyMacro or PrintHandler. In short: mxm.library is the standard MXM system support library. Version 34.14, includes source. Author: Olaf Barthel	Fred Fish Disk 359	
Fred Fish Disk 355		ABridge	An interim solution to Amiga-5 incompatibility problems. Identifies the origin of an Amiga-5 file and modifies it to facilitate easy exchange between Amiga, Videocase, Animation Station, DPaint III, Animation: Editor (v.1.11), The Director, SAAD, Movie2.0, Photon Paint 2.0 and Cel Animator. Fully initialized interface, full AReXX support including a "Find AReXX" option if you start AReXX after running ABridge. This is version 1.0, shareware, binary only. Author: Ron Tarrant, Mythra-nations Animation and Software
Berserker	A viruskiller which checks for certain conditions indicating possible virus infection. Different from other programs of this kind, Berserker does not rely on checksums only, it will also check the possible virus behind the altered checksum. Therefore even new viruses with old infection methods can be traced and resident tools are not touched. Includes source in assembly language. Author: Ralf Thanner	DICE	Dillon's Integrated C Environment. A C frontend, pre-processor, C compiler, assembler, linker, and support libraries. Also includes the editor, dme. Features include ANSI compatibility, many code optimizations, and autoint routines (user routines called during startup before main is called). This is version 2.02, shareware, binary only. Author: Matthew Dillon
ImageEditor	A simple to use graphics editor which allows you to draw and save images/sprites as assembler or C source code. Includes IFF support, undo, and an iconify function. Another feature is the small memory usage so you can use multitasking even on a 512K machine. Maximum picture size is 16658 pixels. This is version 2.4 and includes source. Author: Robert Junghans	TextPlus	A word processor for the Amiga, with both German and English versions. TextPlus enables you to write letters, books, programs etc. in a very easy and comfortable way. Version 2.0, binary only. Author: Martin Steppeler
LoadImage	An IFF ILBM reader that accepts overscanned pictures, allows you to scroll around in the bitmap if the picture is larger than the current display, works on both PAL and NTSC machines, supports color cycling using interrupt code, and supports printing of image portions. Version 1.11, update to version 1.0 on disk 281, includes source. Author: Olaf Barthel	Fred Fish Disk 360	
RexxHostLib	This is a shared library package to simplify the AReXX host creation management procedure. Rexx-message parsing is also included making it possible to control AReXX from programs such as AmigaBASIC (can you imagine AmigaBASIC controlling AmigaTeX?). This is version 34.12 which has been recompiled and made a lot shorter using Aztec C 5.0, an update to version 1.6 on disk 325. Includes source. Author: Olaf Barthel	UUCP	An implementation of uucp for the Amiga, including mail and news. This is Matt's version for the Amiga, based on William Loftus's Amiga UUCP 0.40 release with news code from his 0.60 release, and months of work by Matt to make fixes and add enhancements. This is version 1.050, an update to FF313. Includes source. Author: Various, major enhancements by Matt Dillon
SoundEditor	An 8SVX stereo sound file editor written in assembly language for speed and minimum size. Version V.8, binary only. Author: Howard Dorth, Mike Coriell, Matt Gerald	To Be Continued.....	
TrackSalve	A Trackdisk patch which removes all known bugs, and one unknown so far, and patches the Trackdisk task to allow various enhancements, such as reading good sectors from partially bad tracks, write verification, write protect simulation, auto motor off, auto update and turning off clicking. Other features are MFM-update and I/O by non-chip buffers. This is version 1.3, an update of version 1.0 on disk 312. Includes source in C and assembler. Author: Dirk Reisig	In Conclusion	To the best of our knowledge, the materials in this library are freely distributable. This means they were either publicly posted and placed in the public domain by their authors, or they have restrictions published in their files to which we have adhered. If you become aware of any violation of the authors' wishes, please contact us by mail.
Tron	Another game about the lightcycle race sequence in the science fiction computer film "Tron". One or two players and other options. Written in GFA-BASIC and then compiled. Version 1.1, binary only. Author: Dirk Hasse	IMPORTANT NOTICE!	This list is compiled and published as a service to the Commodore Amiga community for informational purposes only. Its use is restricted to non-commercial purposes only! Any duplication for commercial purposes is strictly forbidden. As a part of Amazing Computing™, this list is inherently copyrighted. Any infringement on this proprietary copyright without expressed written permission of the publishers will incur the full force of legal actions.
Fred Fish Disk 356		Any non-commercial Amiga user group wishing to duplicate this list should contact:	
AlgoRhythms	An algorithmic composition program that improvises music over a MIDI interface connected to the serial port. A MIDI interface and synthesizer are needed. The music does not have a strong pulse, and does not repeat motifs or melodies, but can be very pretty. Version 1.0 with source in C, and sample data files. Author: Thomas E. Janzen	PIM Publications, Inc.	P.O. Box 869 Fall River, MA 02722
NComm	A communications program based on Comm version 1.34, by DJ James, with lots of very nice enhancements. Also includes several auxiliary programs such as AdoCall, CallInfo, GenList, PbConvert, and ReadMail. This is version 1.9, an update to version 1.8 on disk 230. Binary only. Author: DJ James, Daniel Bloch, Torkel Loeberg, et al.	PIM Publications Inc.	is extremely interested in helping any Amiga user groups in non-commercial support for the Amiga.
Fred Fish Disk 357			
Empire	Empire is a multiplayer game of exploration, economics, war, etc., which can last a couple of months. Can be played either on the local keyboard or remotely through a modem. This is version 2.1w, an update to version 1.33w on disk 329. Changes include a client-server system, a chat/GB mode, realtime private player to player messages, and other enhancements. Binary only. Author: Chris Gray, David Wright, Peter Langston		
Fred Fish Disk 358			
Blob	Another screen hack. Makes red drops of slime flow down your screen. Version 1.1, includes source in C. Author: Guido Wegener		

(Snapshot, continued from page 57)

Ship sailing and the dueling segment are the most common ones. They are fairly easy to carry out, though I never did completely grasp what I was doing during the duel, I just mastered an approach that worked.

The graphics are clear and fit in well with the setting, while the digitized sounds are wonderful. The seashore sounds that play during loading almost make you want to look out a nearby window to see if the tide is coming in! While I could single out a few areas for improvement, the game is enjoyable as is.

The four different difficulty levels and selection of playing decades make for a flexible game with a great deal of variety. Several historical missions are also available where you can compete against the records of real captains. I recommend this game highly.

STIK-GRIPPER

One last item. A short while ago, I got a device called a Stik-Gripper, which is designed to hold a joystick firmly in place. Clearly, this is something that can come in handy in many games. I finally put it together last week, and here is my report.

It is a fairly simple design, made out of metal parts that fit together snugly. On the whole, the Stik-Gripper lacks the visual appeal of most modern-day computer products. When the whole thing is put together, the base mounts on the edge of your computer table and the joystick is then mounted in the top section.

I had a bit of a problem attaching the base, since the table I have my Amiga on has a vertical block about an inch-and-a-half in from the edge. The device hooked up fine in the area at the corner of my table where there is no block. The top part accepts nearly any joystick, though I did have some trouble putting one of my largest in (one that already had rubber suction pad feet, so I was not too worried). This is, in fact, a rather useful device. And the price is definitely reasonable.

•AC•

Turbo Outrun

SEGA
1810 Gateway Drive
San Mateo, CA 94404
(415) 571-7171 (to order)
Price: \$49.95
Inquiry #201

Hardball II

Accolade
550 S. Winchester Blvd Ste. 200
San Jose, CA 95128
(408) 985-1700
Price: \$44.95
Inquiry #205

D.R.A.G.O.N. Force

Interstel
1810 Gateway Drive
San Mateo, CA 94404
(415) 571-7171 (to order)
Price: \$49.95
Inquiry #202

Welltris

Spectrum HoloByte
2061 Challenger Drive
Alameda, CA 94501
(415) 522-3584
Price: \$39.95
Inquiry #206

Waterloo

SSI
675 Almanor Ave.
Sunnyvale, CA 94086
(408) 737-6800
Price: \$59.95
Inquiry #203

Pirates

MICROPROSE
180 Lakefront Drive
Hunt Valley, MD 21030
(301) 771-1151
Price: \$44.95
Inquiry #207

Heat Wave

Accolade
550 S. Winchester Blvd Ste. 200
San Jose, CA 95128
(408) 985-1700
Price: \$44.95
Inquiry #204

Stik-Gripper

Duggan DeZign, Inc.
300 Quaker Ln. Ste. #7
Warwick, RI 02886
(800) 843-1223 (to order)
Price: \$18.95
Inquiry #208

Name _____
 Address _____
 City _____ State _____ Zip _____
 Charge my ☐ Visa ☐ MC # _____
 Expiration Date _____ Signature _____



Please circle if this is a **New Subscription** or a **Renewal**
PROPER ADDRESS REQUIRED. In order to expedite and guarantee your order, all large Public Domain Software orders, as well as most Back issue orders, are shipped by United Parcel Service. UPS requires that all packages be addressed to a street address for correct delivery.
PAYMENTS BY CHECK. All payments made by check or money order must be in US funds drawn on a US bank.

All Charges are subject to a \$20.00 minimum (charges under \$20.00 will receive a \$2.00 service charge).

One Year Of Amazing	Save over 49%	<input type="checkbox"/>	\$24.00 U.S.
	12 monthly issues of the number one resource to the Commodore Amiga, Amazing Computing at a savings of over \$23.00 off the newsstand price!	<input type="checkbox"/>	\$44.00 Foreign Surface
		<input type="checkbox"/>	\$34.00 Canada and Mexico
One Year of AC SuperSub!	Save over 47%	<input type="checkbox"/>	\$36.00 U.S.
	12 monthly issues of Amazing Computing PLUS AC's GUIDE / AMIGA	<input type="checkbox"/>	\$64.00 Foreign Surface
	3 Product Guides a year! A savings of \$32.25 off the newsstand price.	<input type="checkbox"/>	\$54.00 Canada and Mexico
Two Years Of Amazing	Save over 59%	<input type="checkbox"/>	\$38.00 US
	24 monthly issues of the number one resource to the Commodore Amiga, Amazing Computing at a savings of over \$56.80 off the newsstand price!		(sorry no foreign orders available at this frequency)
Two Years of AC SuperSub!	Save over 56%	<input type="checkbox"/>	\$59.00 US
	24 monthly issues of Amazing Computing PLUS AC's GUIDE / AMIGA		(sorry no foreign orders available at this frequency)
	6 Complete Product Guides! A savings of \$77.50 off the newsstand price.		

Please circle any additional choices below:

(Domestic and Foreign air mail rates available on request)

Back Issues: \$5.00 each US, \$6.00 each Canada and Mexico, \$7.00 each Foreign Surface.

1.1 1.2 1.3 1.4 1.5 1.6 1.7 1.8 1.9 2.1 2.2 2.3 2.4 2.5 2.6 2.7 2.8 2.9 2.10
 2.11 2.12 3.1 3.2 3.3 3.4 3.5 3.6 3.7 3.8 3.9 3.10 3.11 3.12 4.1 4.2 4.3 4.4 4.5
 4.6 4.7 4.8 4.9 4.10 4.11 4.12 5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8

Back Issue Volumes: Volume 1-\$19.95* Volume 2-\$29.95* Volume 3-\$29.95* Volume 4-\$29.95*

*All volume orders must include postage and handling charges: \$4.00 each set US, \$7.50 each set Canada and Mexico, and \$10.00 each set for foreign surface orders. Airmail rates available.

Freely Distributable Software:

Subscriber Special (yes, even the new ones!)

1 to 9 disks \$6.00 each
 10 to 49 disks \$5.00 each
 50 to 100 disk \$4.00 each
 100 or more disks \$3.00 each

\$7.00 each for non subscribers (three disk minimum on all foreign orders)

Amazing on Disk: AC#1...Source & Listings V3.8 & V3.9

AC#3...Source & Listings V4.5 & V4.6

AC#5...Source & Listings V4.9

AC#7...Source & Listings V4.12 & V5.1

AC#9...Source & Listings V5.4 & V5.5

AC#2...Source & Listings V4.4

AC#4...Source & Listings V4.7 & V4.8

AC#6...Source & Listings V4.10 & V4.11

AC#8...Source & Listings V5.2 & 5.3

AC#10...Source & Listings V5.6 & 5.7

AC#11...Source & Listings V5.8 & 5.9

InNOCKulation Disk: IN#1...Virus protection

AMICUS 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19
 20 21 22 23 24 25 26

Fred Fish Disks

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50
 51 52 53 54 55 56 NA 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75
 76 77 78 79 NA 81 82 83 84 85 86 NA 88 89 90 91 92 93 94 95 96 97 98 99 100
 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150
 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175
 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200
 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225
 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250
 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275
 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300
 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325
 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350
 351 352 353 354 355 356 357 358 359 360

(NA Denotes disks removed from the collection)

Subscription: \$ _____

Back Issues: \$ _____

PDS Disks: \$ _____

Total: \$ _____

Please complete this form and mail with check, money order or credit card information to:

PIM Publications, Inc.

P.O. Box 869

Fall River, MA 02722-0869

Please allow 4 to 6 weeks for delivery

Complete Today, or Telephone 1-800-345-3360

THE NEW AmiEXPO

THE

AMIGA

PERSONAL COMPUTER SHOW

October 5-7 at The Disneyland Hotel Anaheim, California

Sponsored by

AMIGA

WORLD

**Admission includes the Exhibition, Seminars, Keynotes &
Amiga Artists Theatre!**

**120 Amiga Exhibitors Featuring State of the Art
Software and Hardware, at the lowest prices!**

**Master Classes Available in Amiga Graphics, Video, MultiMedia, Animation, Rendering and Publishing!
Seating for Master Classes is limited; call for schedule and availability before registering.**

PRE-REGISTRATION DEADLINE IS SEPTEMBER 21, 1990

(No cancellations or refunds after deadline)

For Hotel Reservations Call the Disneyland Hotel at (714) 778-6600

Hotel reservations deadline: September 19, 1990

For discounted airfares, call American Airlines at (800) 433-1790 and give them this ID: 12Z 04F

REGISTER TODAY!

**Register by Mail using the coupon below or Call 800-32-AMIGA Nationwide (or 914-741-6500)
For Your Ticket to The Amiga Event!**

Yes, I want to come to AmiEXPO-California

☐ Friday ☐ Saturday ☐ Sunday

**Registration is
\$5 Additional
At The Door**

One day - \$15 ☐

Two days - \$20 ☐

Three days - \$25 ☐

Master Class(es) - List Class and Time - \$60 Each

NFAW

Total Amount Enclosed _____

NAME _____

COMPANY _____

ADDRESS _____

CITY _____ STATE _____ ZIP _____

For ☐ MasterCard or ☐ VISA Payment

Expiration Date _____

Account Number _____

Name as it appears on card: _____

Signature _____

Make Check or Money Order Payable to:
**AmiEXPO 465 Columbus Ave., Ste. 285
Valhalla, NY 10595**

Title PageTM

Title Page is a new video titling package for the Amiga. It will finally allow you to create screens full of effects possible once only in your imagination! If the 'look' you want is not in our package, simply create it! Modify text, effects, patterns, brushes, even backgrounds. If that's not enough, add a touch of fantasy with rainbow letters. So if what you use isn't what you need, come experience Title Page.

- Supports all video modes, except HAM
- User selectable overscan level
- Create copper display lists allowing thousands of extra colors per screen



- Use over 65 **ARexx** commands to customize **Title Page** to your needs
- Includes an **ARexx** compatible slideshow player with 45 different transitions



Joe's Classic Leasing, Inc.

I'll get you the best car you can afford!

- Apply 40+ effects to text, brushes, or images
- Use standard Amiga fonts.
- Includes 9 regular fonts in 3 sizes PLUS 4 colorfonts in 2 sizes.
- Keymap support allows you to use **accents**



Title Page functions properly on any 512Kb Amiga. We also remembered that everyone's needs are not the same, so we included a variety of features for users with more memory and faster CPUs.

Only \$199.95

ESCHALON DEVELOPMENT

For more information call:
(514) 340-9244

Circle 125 on Reader Service card.